计算数学丛书

外推法及其应用

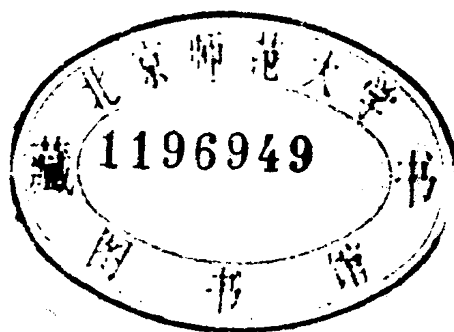
邓建中

上海科学技术出版社

计算数学丛书

外推法及其应用

邓 建 中



上海科学技术出版社

计算数学丛书

外推法及其应用

邓 建 中

上海科学技术出版社出版

（上海瑞金二路 450 号）

新华书店上海发行所发行 **无锡县人民印刷厂印刷**

开本 787×1092 1/32 印张 5.25 字数 114,000

1984年 3月第1版 1984年 3月第1次印刷

印数：1—8,200

统一书号：13119·1124 定价：（科五）0.63 元

出 版 说 明

《计算数学丛书》是为了适应计算数学和计算机科学的发展，配合高等院校计算数学教学的需要而组织的一套参考读物。读者对象主要是高等院校数学系和计算机科学系的学生、研究生，亦可供高等院校数学系和计算机科学系的教师以及工矿企业、科研单位从事计算工作的技术人员参考。

本丛书向读者介绍近代计算方法的一些主要进展及其适用范围和实用效果。每种书集中介绍一个专题，针对本专题的近代发展作综合性的介绍，内容简明扼要，重点突出，有分析，有评价，力图使读者对该专题的动向和发展趋势得到一个完整的了解。

本丛书已拟定的选题计有：《线性代数与多项式的快速算法》、《数论变换》、《数值有理逼近》、《矩阵特征值问题》、《索伯列夫空间引论》、《计算组合数学》、《样条与插值》、《有限条形法》、《广义逆矩阵及其计算方法》、《非线性方程迭代解法》、《奇异摄动中的边界层校正法》、《沃尔什函数理论与应用》、《多项式最佳逼近的实现》、《坏条件常微分方程数值解》、《误差分析》、《最小二乘问题的数值解法》、《板壳问题非协调方法》、《外推法及其应用》、《Monte Carlo 方法》、《差分格式理论》、《高维偏微分方程数值解》等二十余种，于一九八〇年初起陆续出版。

引　言

外推法，即外推极限法(Extrapolation to the limit)或延伸趋限法(Deferred approach to the limit)，是最近二十年迅速发展的一种计算方法，目前已应用到数值计算的各个方面．数值积分中广泛使用的 Romberg 积分法，常微分方程数值解法中通常最有效的 GBS 算法，就是外推法．1971 年 D. J. Joyce 的综合报告，全面、系统地叙述了外推法的发展情况，是一篇难得的好文章．

本书的目的，是以 Joyce 的报告为线索，参照近年来发表的文献，结合作者学习外推法的体会，对外推法的理论基础及应用，作比较系统的介绍．全书共分三部分．第一部分，即第 1 章，以 Romberg 外推算法为例，介绍了外推法的一般概念、计算步骤和历史．第二部分，即第 2、3、4 章，分别对多项式外推法、有理式外推法和 ε 算法的计算格式，作了比较详细的推导；并讨论了格式的收敛性与稳定性．第三部分，即第 5、6、7 章，介绍了外推法的应用．由于作者水平有限，资料缺乏，难免有不少错误与不足，敬请读者批评指正．

在本书编写过程中，西安交通大学游兆永教授和上海科学技术出版社理科编辑室，给予了热情的鼓励和帮助．借此机会，作者谨向他们表示衷心的感谢．

邓建中

于西安交通大学

1980.5.

目　录

引言
第1章　什么叫外推法 ……………………………………1
　§1　数学史上的一个疑案 ……………………………………1
　§2　Romberg 外推算法 ……………………………………2
　§3　Romberg 算法的理论根据 ……………………………………5
　§4　外推算法在计算机上的实现 ……………………………………8
　§5　外推法研究的问题 ……………………………………11
第2章　多项式外推法 ……………………………………12
　§1　多项式插值法基础 ……………………………………12
　§2　多项式外推法及其推广 ……………………………………16
　§3　广义多项式外推法 ……………………………………24
　§4　广义多项式插值法与 Richardson 外推法的关系 ……………………………………31
　§5　误差估计 ……………………………………33
　§6　收敛性与稳定性 ……………………………………39
第3章　有理式外推法 ……………………………………45
　§1　有理式插值的概念 ……………………………………45
　§2　连分式算法 ……………………………………46
　§3　Larkin 逐步插值法 ……………………………………50
　§4　有理式逐步外推法 ……………………………………56
　§5　误差估计 ……………………………………59
　§6　收敛性与稳定性 ……………………………………63
第4章　ε 算法 ……………………………………66
　§1　ε 算法的导出 ……………………………………66
　§2　ε 算法的若干性质 ……………………………………76

第 5 章　外推法的应用(一)：数值积分与微分……84
§ 1　Euler–Maclaurin 求和公式……84
§ 2　一些简单求积公式的导出……93
§ 3　Romberg 积分法及其改进……98
§ 4　反常积分……103
§ 5　重积分……109
§ 6　数值微分……111
第 6 章　外推法的应用(二)：函数方程数值解……115
§ 1　Euler 折线法与外推……115
§ 2　离散化方法与外推……124
§ 3　全程误差渐近展开式……130
§ 4　GBS 算法……138
第 7 章　外推法的应用(三)：其它……144
§ 1　加速序列的收敛……144
§ 2　级数求和与特殊函数的求值……149
§ 3　方程求根……150
§ 4　代数方程组的求解……152
参考文献……157
算法索引……161

第 1 章

什么叫外推法

§1 数学史上的一个疑案

为了说明什么叫外推法，我们先从数学史上的一个疑案谈起.

《隋书·律历志》记载，我国南北朝时代数学家祖冲之(公元 429—500 年)算出了圆周率

$$3.1415926<\pi<3.1415927.$$

这是一个光辉的成就. 这样精确的结果，比西方早了一千多年，直到十五世纪中期阿拉伯数学家 al-Kashi 和十六世纪法国数学家 Vieta 才打破了这一记录. 祖冲之是怎样算出来的呢？由于古书失传，现在还未考证清楚.

毫无疑问，祖冲之一定会受到魏晋时代数学家刘徽的影响. 刘徽在公元 263 年注的《九章算术》中，算出了当时世界上最精确的圆周率 $\pi=3.1416$. 他的方法是从半径为 10 的圆内接正六边形出发，逐步算出正 12、24、48、96 和192 边形的面积. 他指出，正多边形的边数越多，其面积就越接近于圆的面积 100π. 值得注意的是，他算出 $\pi=3.1416$，并不是计算 3072 边形得出来的. 他的算法是：先算出圆内接正 96、192 边形的面积

$$S(96)=313\frac{584}{625},\quad S(192)=314\frac{64}{625},$$

求出两者的差

$$S(192)-S(96)=\frac{105}{625},$$

然后"以十二觚之幂为率消息","取此分寸之三十六",即 $\frac{36}{625}$, 加到 $S(192)$, 得

$$100\pi \approx 314\frac{64}{625}+\frac{36}{625}=314\frac{4}{25}=314.16. \quad (1.1)$$

这里引号内的话,是刘徽的原话,现在怎样解释,则众说纷纭,没有定论. 不过从他的计算过程来看,他是利用了某种方法,根据几个内接正多边形的面积,来推算这种多边形边数无限增加时面积的极限值. 这"某种方法",可能就是现在的 Romberg 外推法;而祖冲之算出他的圆周率,也可能是现在的 Romberg 外推法.

§2 Romberg 外推算法

利用 Romberg 外推算法,可以根据圆内接正二、三、四、六、八边形的周长,算出跟祖冲之差不多的结果. 事实上,设圆的直径为 1, 记其内接正 n 边形的周长为 $L(n)$, 则

$$L(2)=2,$$

$$L(3)=\frac{\sqrt{9}}{2}\approx 2.598076211,$$

$$L(4)=\sqrt{8}\approx 2.828427125,$$

$$L(6)=3,$$

$$L(8)=4\sqrt{2-\sqrt{2}}\approx 3.061467459.$$

设 $h=\frac{1}{n}$, $T(h)=L(n)$, 令

$$T_0^{(i)}=T(h_i),\ i=0,\ 1,\ 2,\ \cdots, \quad (1.2)$$

$$T_m^{(i)}=\frac{h_i^2 T_{m-1}^{(i+1)}-h_{i+m}^2 T_{m-1}^{(i)}}{h_i^2-h_{i+m}^2}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{(h_i/h_{i+m})^2-1},\tag{1.3}$$

则按表 1 可得表 2.

表 1 外推表的一般形式

$T_0^{(0)}$				
$T_0^{(1)}$	$T_1^{(0)}$			
$T_0^{(2)}$	$T_1^{(1)}$	$T_2^{(0)}$		
$T_0^{(3)}$	$T_1^{(2)}$	$T_2^{(1)}$	$T_3^{(0)}$	
$T_0^{(4)}$	$T_1^{(3)}$	$T_2^{(2)}$	$T_3^{(1)}$	$T_4^{(0)}$
⋮	⋮	⋮	⋮	⋮

表 2 计算 π 的外推表

2.000000000				
2.589076211	3.076537180			
2.828427125	3.124592586	3.140611055		
3.000000000	3.137258300	3.141480205	3.141588849	
3.061467459	3.140497049	3.141576632	3.141592411	3.141592648

这里 $T_m^{(i)}$ 为表中第 m 列的第 i 行元素，而 m 和 i 的编号从 0 算起. 例如表 2 中

$$\begin{aligned}T_2^{(2)}=3.141576632&=T_1^{(3)}+\frac{T_1^{(3)}-T_1^{(2)}}{(h_2/h_4)^2-1}\\&=3.140497049\\&\quad+\frac{3.140497049-3.137258300}{\left(\frac{1}{4}\Big/\frac{1}{8}\right)^2-1}.\end{aligned}$$

从表 2 可见，表中各列 $(T_m^{(0)}, T_m^{(1)}, T_m^{(2)}, \cdots)$、各行 $(T_0^{(i)}, T_1^{(i-1)}, T_2^{(i-2)}, \cdots, T_i^{(0)})$ 以及各对角线 $(T_0^{(i)}, T_1^{(i)}, T_2^{(i)}, \cdots)$ 上的元素，都单调地趋于 $\{T(h_i)\}$ 的极限值 π；且右列比左列，下行比上

行，下一对角线比上一对角线，收敛速度越来越快；$T_4^{(0)}=3.141592648$ 相当于祖冲之的圆周率近似值.

像表 2 那样，利用公式(1.2)和(1.3)，按照表 1 计算 $T_m^{(i)}$，用来近似 $T(h)$ 当 $h\to 0$ 时极限值 τ_0 的方法，称为 **Romberg 外推算法**. $T(h)$ 的极限值 τ_0 往往不可能通过有限次算术运算求出来；但对 h 的某些离散数值 h_i, $h_i\neq 0$, $T(h_i)$ 却常常可通过有限次、比较复杂的过程计算出来，而且可用来近似 τ_0，只是误差较大. 当 $m>0$ 时，由(1.3)可见，$T_m^{(i)}$ 是 $T_{m-1}^{(i+1)}$ 与 $T_{m-1}^{(i)}$ 的线性组合，而 $T_{m-1}^{(i+1)}$ 与 $T_{m-1}^{(i)}$ 又分别是 $T_{m-2}^{(i+2)}$、$T_{m-2}^{(i+1)}$ 与 $T_{m-2}^{(i+1)}$、$T_{m-2}^{(i)}$ 的线性组合，……，如此可知 $T_m^{(i)}$ 是 $T_0^{(i)}$, $T_0^{(i+1)}$, …, $T_0^{(i+m)}$ 的线性组合，或 $T(h_i)$, $T(h_{i+1})$, …, $T(h_{i+m})$ 的线性组合. $T_m^{(i)}$ 近似 τ_0 的误差往往比 $T(h_i)$、$T(h_{i+1})$、…、$T(h_{i+m})$ 的都小. 因此，Romberg 外推法，实质上是通过 τ_0 的若干离散化近似值 $T(h_k)$ 的适当组合，来推算极限值 τ_0，即用 $T_m^{(i)}$ 来近似 τ_0.

如果把 Romberg 外推法应用于刘徽的数据，令 $h_0=\frac{1}{96}$，$h_1=\frac{1}{192}$，$T(h_0)=313\frac{584}{625}$，$T(h_1)=314\frac{64}{625}$，则按(1.2)、(1.3)有

$$T_0^{(0)}=313\frac{584}{625},\qquad T_0^{(1)}=314\frac{64}{625},$$

$$\begin{aligned}T_1^{(0)}&=T_0^{(1)}+\frac{T_0^{(1)}-T_0^{(0)}}{(h_0/h_1)^2-1}\\&=314\frac{64}{625}+\frac{314\frac{64}{625}-313\frac{584}{625}}{\left(\frac{1}{96}\Big/\frac{1}{192}\right)^2-1}\\&=314\frac{64}{625}+\frac{105/625}{3}=314\frac{64}{625}+\frac{35}{625}.\end{aligned}\tag{1.4}$$

比较(1.1)，可见这里的修正项 $\frac{35}{625}$ 跟刘徽的修正项 $\frac{36}{625}$ 十分接近，这说明刘徽有可能使用 Romberg 外推法，按公式(1.3)算了一步．祖冲之生在刘徽之后一百多年，很可能进一步使用 Romberg 外推法，按公式(1.3)算了好几步．

然而刘徽的修正项毕竟与 Romberg 外推法的修正项不同．也许刘徽不是用的公式(1.3)，而是用的别的公式．但无论如何，他总是利用若干 $T(h_k)$ 的某种组合，来推算序列 $\{T(h_k)\}$ 的极限 τ_0 的精确近似值，只是计算 $T_m^{(i)}$ 不一定按公式(1.3)而已．现代各种各样的外推法，其实也是利用

$$T(h_i)、T(h_{i+1})、\cdots、T(h_{i+m})$$

的适当组合来推算极限 τ_0 的精确近似值，只是采用了各种各样的公式来计算 $T_m^{(i)}$．所以外推法全称**外推极限法**，或称**延伸趋限法**，又称**加速收敛法**．看来刘徽或祖冲之也可能采用了另外一种外推法．当然，这些都只是猜测，实际情况究竟如何，还需留待考古新发现和数学史专家的考证．

§3 Romberg 算法的理论根据

前面我们看到，从 π 的几个粗略近似值出发，利用 Romberg 算法便能得出 π 的精确近似值．这是什么道理呢？下面来回答这个问题．

设圆的直径为 1，则其内接正 n 边形的周长

$$L(n)=n\sin\frac{\pi}{n}.$$

按照 Taylor 公式

$$L(n)=n\left[\frac{\pi}{n}-\frac{1}{3!}\left(\frac{\pi}{n}\right)^3+\frac{1}{5!}\left(\frac{\pi}{n}\right)^5-\frac{1}{7!}\left(\frac{\pi}{n}\right)^7+\cdots\right]$$

$$=\pi-\frac{\pi^3}{3!}\left(\frac{1}{n}\right)^2+\frac{\pi^5}{5!}\left(\frac{1}{n}\right)^4-\frac{n^7}{7!}\left(\frac{1}{n}\right)^6+\cdots.$$

令 $h=\frac{1}{n}$, $T(h)=L(n)$, $\tau_k=(-1)^k\pi^{2k+1}/(2k+1)!$, $k=1$, 2, …, 则

$$T(h)=\pi+\tau_1h^2+\tau_2h^4+\tau_3h^6+\cdots. \tag{1.5}$$

于是

$$T(h_i)-\pi=\tau_1h_i^2+\tau_2h_i^4+\tau_3h_i^6+\cdots, \tag{1.6}$$

$$T(h_{i+1})-\pi=\tau_1h_{i+1}^2+\tau_2h_{i+1}^4+\tau_3h_{i+1}^6+\cdots. \tag{1.7}$$

这里 $T(h_i)-\pi=L(n_i)-\pi$ 是用圆内接正 n_i 边形的周长近似 π 的误差，即用离散近似值 $T(h_i)$ 近似极限值 $\tau_0=\pi$ 的误差，称为**离散化误差**. 公式(1.6)称为 $T(h_i)$ 的**离散化误差渐近展开式**. 由于外推表中 $T_0^{(i)}=T(h_i)$, 公式(1.6)表示, 用外推表中 0 列元素 $T_0^{(i)}$ 近似 π, 其误差为 $O(h_i^2)$.

将(1.6)、(1.7)分别乘 h_{i+1}^2、h_i^2, 然后相减, 除以 $h_{i+1}^2-h_i^2$, 并注意公式(1.3), 则得

$$\begin{aligned}T_1^{(i)}-\pi&=\frac{h_i^2T_0^{(i+1)}-h_{i+1}^2T_0^{(i)}}{h_i^2-h_{i+1}^2}-\pi\\&=-h_i^2h_{i+1}^2[\tau_2+\tau_3(h_i^2+h_{i+1}^2)\\&\quad+\tau_4(h_i^4+h_i^2h_{i+1}^2+h_{i+1}^4)\\&\quad+\tau_5(h_i^6+h_i^4h_{i+1}^2+h_i^2h_{i+1}^4+h_{i+1}^6)+\cdots].\end{aligned}$$

这说明, 用外推表中第 1 列元素 $T_1^{(i)}$ 近似 π, 其误差比 0 列小, 为 $O(h_i^2h_{i+1}^2)$. 类似地可得

$$\begin{aligned}T_2^{(i)}-\pi&=\frac{h_i^2T_1^{(i+1)}-h_{i+2}^2T_1^{(i)}}{h_i^2-h_{i+2}^2}-\pi\\&=h_i^2h_{i+1}^2h_{i+2}^2[\tau_3+\tau_4(h_i^2+h_{i+1}^2+h_{i+2}^2)\\&\quad+\tau_5(h_i^4+h_i^2h_{i+1}^2+h_i^2h_{i+2}^2+h_{i+1}^2h_{i+2}^2+h_{i+1}^4+h_{i+2}^4)+\cdots],\end{aligned}$$

$$T_3^{(i)}-\pi=\frac{h_i^2T_2^{(i+1)}-h_{i+3}^2T_2^{(i)}}{h_i^2-h_{i+3}^2}-\pi$$
$$=-h_i^2h_{i+1}^2h_{i+2}^2h_{i+3}^2[\tau_4+\tau_5$$
$$\times(h_i^2+h_{i+1}^2+h_{i+2}^2+h_{i+3}^2)+\cdots],$$
$$T_4^{(i)}-\pi=\frac{h_i^2T_3^{(i+1)}-h_{i+4}^2T_3^{(i)}}{h_i^2-h_{i+4}^2}-\pi$$
$$=\pi+h_i^2h_{i+1}^2h_{i+2}^2h_{i+3}^2h_{i+4}^2$$
$$\times[\tau_5+\tau_6(h_i^2+h_{i+1}^2+h_{i+2}^2+h_{i+3}^2+h_{i+4}^2)+\cdots].$$
(1.8)

这些式子表明，外推表中的列号越大，即 $T_m^{(i)}$ 的 m 越大，所含误差的阶数越高. 因此，Romberg 算法的计算过程，实质上是逐列消去低次误差项、逐列提高误差阶数的过程，从而能逐步地使所得近似值越来越精确.

由公式(1.8)还可知，在表 2 中

$$3.141592648-\pi=T_4^{(1)}-\pi$$
$$\approx-\frac{\pi^{11}}{11!}\left(\frac{1}{2}\right)^2\left(\frac{1}{3}\right)^2\left(\frac{1}{4}\right)^2\left(\frac{1}{6}\right)^2\left(\frac{1}{8}\right)^2$$
$$\approx-0.55\times10^{-8}.$$

π 准确到小数点后十位数字的近似值为 3.1415926535，它同 3.141592648 的差恰好是 0.55×10^{-8}，可见这里的误差估计是很精确的.

从上面公式的推导来看，如果把 π 改为另外某个量 τ_0，它是 $T(h)$ 当 $h\to0$ 时的极限值，而 $T(h)$ 具有形如(1.5)的渐近展开式，即

$$T(h)=\tau_0+\tau_1h^2+\tau_2h^4+\tau_3h^6+\cdots+\tau_Nh^{2N}+\tau_{N+1}(h)h^{2N+2},\tag{1.9}$$

那么把 Romberg 外推法(1.3)应用于离散化近似值 $\{T(h_i)\}$，同样能逐列消去误差项，使所得近似值越来越精确. 正是由

于这个原因，Romberg 外推法有着广泛的应用．Romberg 积分法就 Romberg 外推法在数值积分的应用．在那里，需求的极限值 τ_0 是积分 $\int_a^b f(x)dx$，$T(h)$ 是将 $[a, b]$ 分成长 h 的小区间时，按梯形求积公式算出的积分近似值，而 $h_0=b-a$，$h_i=h_0 2^{-i}(i=1, 2, \cdots)$．

上述逐步消去近似值中误差项的思想，是 L. F. Richardson 1910 年提出来的，因而常称为 **Richardson 外推法**．其实，他提出这算法的时候还不够一般化，只是通过二、三个离散化近似值消去低阶误差项，来推算精确近似值，即只按公式(1.3)计算外推表中的一、二列元素 $T_1^{(i)}$ 和 $T_2^{(i)}$，并称之为 **h^2 外推**、**(h^2、h^4)外推**．他的这种思想许多人早就用过，如 Huygens(1654)、Sheppard(1900)、Milne(1903) 以及刘徽．用递推公式(1.3)不仅可以消去含 h^2、h^4 的误差项，而且还可以逐步消去含 h^6、h^8、… 等的误差项，这是 1955 年 Romberg 首先看出来的．Richardson 的功劳，在于他明确提出了 h^2 外推与(h^2、h^4)外推；并且应用到多种问题，获得了成功．

§4 外推算法在计算机上的实现

在应用外推法(不仅 Romberg 算法)时，离散化参数数列 $\{h_i\}$ 通常取作单调趋于 0 的正数列，即

$$h_0>h_1>h_2>\cdots>0 \quad 且 \quad \lim_{i\to\infty} h_i=0.$$

在正常情况下，由于 $T_m^{(i)}$ 的首项误差 $(-1)^m\tau_{m+1}h_i^2h_{i+1}^2\cdots h_{i+m}^2$(参见(1.6)～(1.8))在误差中起主要作用，当 i 增大时，$\{T_m^{(i)}\}$ 会单调地趋于极限值 τ_0，即外推表中每列的元素会单调地趋于极限值 τ_0．每一行上的元素，由于逐列消去低次误

差项的结果，自然会更快地趋于极限值 τ_0. 这样，在只能用有限位数字表示数的计算机上，经过若干计算之后，可能会很快得到 τ_0 的足够精确的近似值，进一步计算的意义反而会被舍入误差的影响所淹没. 因此，通常一旦发现同行或同列相邻二元素相差很小时，或者外推的步数和次数(即外推表的列数和行数)太大时，例如超过事先规定的数 M_1、M_2 时，便停止计算. 在尾数为 8 位、12 位的机器上，M_1 分别取 6、10 较为适宜. 如果超过 M_1 或 M_2 仍不能求得满意的近似值，说明要求太苛刻，或者用过的 h_0、h_1、$\cdots$ 太大，或者形为(1.9)的展开式不存在，或者 m 超过了(1.9)中的 N.

在许多情况下，h_i 越小，离散化近似值 $T(h_i)$ 的计算工作量也越大. 为了少算 $T(h_i)$ 多作外推，常常采取如下计算顺序：

$$
\begin{array}{l}
\quad T_0^{(0)}\rightarrow \\
\rightarrow T_0^{(1)}\rightarrow T_1^{(0)}\rightarrow \\
\rightarrow T_0^{(2)}\rightarrow T_1^{(1)}\rightarrow T_2^{(0)}\rightarrow \\
\rightarrow T_0^{(3)}\rightarrow T_1^{(2)}\rightarrow T_2^{(1)}\rightarrow T_3^{(0)}\rightarrow \\
\quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\rightarrow T_0^{(i)}\rightarrow T_1^{(i-1)}\rightarrow T_2^{(i-2)}\rightarrow\cdots\rightarrow T_i^{(0)}\ (\text{或 } T_{M_1}^{(i-M_1)}) \\
\rightarrow T_0^{(i+1)}\rightarrow T_1^{(i)}\rightarrow T_2^{(i-1)}\rightarrow\cdots\cdots
\end{array}
$$

这就是说，外推表常常采用一行一行地逐行构造的方法. 按照这种顺序计算，每行元素的误差阶数逐列提高，直到最后元素 $T_i^{(0)}$ 或 $T_{M_1}^{(i-M_1)}$ 仍不能满足误差要求时，才转入下行，缩小 h_i，另算 $T(h_i)$，继续外推. 这样既变阶又缩小 h_i，往往能很快满足误差要求，停止多余的计算. 但为了防止同列元素相差不大但仍远离极限值的现象，最好规定行数的下限，如 m_3.

在逐行构造外推表的过程中，任一元素 $T_m^{(i)}$ 算出之后，上

一行的前列元素 $T_{m-1}^{(i)}$ 一般不再使用. 因此, 可用一个一维数组如 $A[0:M]$ 来存放同一行的元素, 算出 $T_m^{(i)}$ 后将原来存放 $T_{m-1}^{(i)}$ 的地方改存 $T_{m-1}^{(i+1)}$.

根据以上说明, Romberg 外推法可按如下"程序"执行:

1. 算 $T(h_0)\Rightarrow A[0]$;

2. 对 $i=1\sim M_2$ 做:

(1) 算 $T(h_i)\Rightarrow D_1$;

(2) 对 $m=1\sim\min(i, M_1)$ 做:

$$(D_1-A[m-1])/((h_{i-m}/h_i)^2-1)\Rightarrow D_0,$$

$$D_1\Rightarrow A[m-1],\quad D_0+D_1\Rightarrow D_1;$$

(3) $D_1\Rightarrow A[\min(i, M_1)]$;

(4) 当 $i>m_3$ 时检查 $|D_0|$ 是否很小? 若是, 打印结果 D_1, 转 4;

3. 打印失败标志;

4. 停止.

由此可见, 外推算法的程序是比较简单的.

在实际应用中, 参数数列 $\{h_i\}$ 常常使用以下三种:

H_1: $h_i=h_0b^i\left(0<b<1, \text{ 通常 } b=\dfrac{1}{2}\right)$;

H_2: $\{h_i\}=\left\{h_0, \dfrac{h_0}{2}, \dfrac{h_0}{3}, \dfrac{h_0}{4}, \dfrac{h_0}{6}, \dfrac{h_0}{8}, \dfrac{h_0}{12}, \cdots\right\}$, 即 i 为奇数时 $h_i=h_0/2^{\frac{i+1}{2}}$, i 为偶数时 $h_i=h_0/(3\times2^{\frac{i}{2}-1})$;

H_3: $h_i=h_0/(i+1)$.

使用第一种数列 H_1 时 h_i 按比例缩小, $T(h_i)$ 往往好算一些. 例如要算 π, 按刘徽的办法取 $n_0=6$, $n_1=12$, $n_2=24$, $\cdots$, 即取

$$h_0=\frac{1}{6},\quad h_i=h_0/2^i,$$

则$T(h_0)=3$(内接正六边形周长)，而$T(h_{i+1})$可由$T(h_i)$按下式算出：

$$T(h_{i+1})=\frac{\sqrt{2}\,T(h_i)}{\sqrt{1+\sqrt{1-h_i^2T^2(h_i)}}}.$$

这公式是根据勾股定理推出来的．如果内接正多边形的边数不是二倍地增加，则不可能推出这公式．一般情况下，h_i越小，计算$T(h_i)$的工作量越大．此时，如果使用第一种数列H_1，由于h_i缩小较快，计算$T(h_i)$的工作量也就增加较快，因此可以改用第三种数列H_3．不过，由于H_3的参数h_i缩小太慢，计算中舍入误差的影响较大；而且在计算$T(h_i)$时往往难以利用以前算出的数据，所以最好还是应用第二种数列H_2．注意使用H_2不如使用H_1的程序简单．

§5　外推法研究的问题

Romberg外推算法计算简单，收敛快，能够用较少的工作量获得最精确的结果，而且适于变阶、缩小离散化参数、自动提高精确度，因而受到普遍的重视．人们自然要问：在数值分析的其它问题中，对于某种离散化过程，这种算法是否也能用？由于Romberg算法实际上是逐步消去(1.9)那类离散化近似值的低阶误差项，这就必须研究该种离散化过程是否具有(1.9)形式的渐近展开式？如果不具有这种形式的渐近展开式，能否将离散化过程加以改造，使具有一定形式的渐近展开式？或者是否还有新的、更好、更一般的外推算法，能加速离散化过程的收敛？对于一种算法，它的理论根据、应用范围、误差估计、收敛性、稳定性等等，自然必须研究．这些问题也就构成外推法的研究内容．

第 2 章

多项式外推法

§1 多项式插值法基础

Romberg 外推算法求出的 $T_m^{(i)}$，其实不仅可看作 $T(h_i)$ 逐步消去误差项的结果，也可看作 $T(h)$ 的插值多项式当 $h=0$ 时的值．为了说明这一点，也为了进一步推广，我们先回忆一下多项式插值法的基本知识．

大家知道，所谓函数 $f(x)$ 的插值多项式 $P_m^{(i)}(x)$，就是在 $m+1$ 个节点 $x_i, x_{i+1}, \cdots, x_{i+m}$ 处满足插值条件

$$P_m^{(i)}(x_k)=f(x_k) \quad (k=i, i+1, \cdots, i+m) \tag{2.1}$$

的次数至多为 m 的多项式．这里假定节点互不相同．设

$$P_m^{(i)}(x)=a_0+a_1x+a_2x^2+\cdots+a_mx^m, \tag{2.2}$$

则插值条件(2.1)相当于要求系数 $a_0, a_1, \cdots, a_m$ 满足方程组

$$a_0+a_1x_k+a_2x_k^2+\cdots+a_mx_k^m=f(x_k) \quad (k=i, i+1, \cdots, i+m). \tag{2.3}$$

由于这方程组的系数行列式(Vandermonde 行列式)

$$V(x_i, x_{i+1}, \cdots, x_{i+m})=\begin{vmatrix} 1 & x_i & x_i^2 & \cdots & x_i^m \\ 1 & x_{i+1} & x_{i+1}^2 & \cdots & x_{i+1}^m \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{i+m} & x_{i+m}^2 & \cdots & x_{i+m}^m \end{vmatrix} = \prod_{k=i+1}^{i+m}\prod_{j=i}^{k-1}(x_k-x_j)\neq 0, \tag{2.4}$$

它的解存在而且唯一．这说明，满足插值条件(2.1)的次数至

多为 m 的插值多项式存在而且唯一，同时与节点的顺序无关.

插值多项式可以有不同的表达形式.

一种，是 Lagrange 插值公式

$$
\begin{aligned}
P_m^{(i)}(x) &= \sum_{k=i}^{i+m} L_k(x) f(x_k) \\
&= \sum_{k=i}^{i+m} \left[\prod_{j=i}^{i+m}{}' \frac{x-x_j}{x_k-x_j} \right] f(x_k),
\end{aligned}
\qquad (2.5)
$$

其中 $\prod'$ 表示 $j \neq k$ 的乘积. 这公式表出了插值多项式为节点处函数值线性组合的特点，但因"组合系数"$L_k(x)$在节点变化时需从头计算，它不便于改变节点和次数.

另一种，是 Newton 插值公式

$$
\begin{aligned}
P_m^{(i)}(x) = C_0^{(i)} &+ C_1^{(i)}(x-x_i) + C_2^{(i)}(x-x_i)(x-x_{i+1}) + \cdots \\
&+ C_m^{(i)}(x-x_i)(x-x_{i+1})\cdots(x-x_{i+m-1}).
\end{aligned}
\qquad (2.6)
$$

它也可以改写为如下"链锁"的形式：

$$
\left\{
\begin{aligned}
P_m^{(i)}(x) = V_0^{(i)}(x) &= C_0^{(i)} + (x-x_i)V_1^{(i)}(x), \\
V_1^{(i)}(x) &= C_1^{(i)} + (x-x_{i+1})V_2^{(i)}(x), \\
V_2^{(i)}(x) &= C_2^{(i)} + (x-x_{i+2})V_3^{(i)}(x), \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
V_{m-1}^{(i)}(x) &= C_{m-1}^{(i)} + (x-x_{i+m-1})V_m^{(i)}(x), \\
V_m^{(i)}(x) &= C_m^{(i)}.
\end{aligned}
\right.
\qquad (2.7)
$$

令 $x = x_{i+k}$，注意 $P_m^{(i)}(x_{i+k}) = f(x_{i+k})$，则有

$$
\begin{aligned}
&V_0^{(i)}(x_{i+k}) = f(x_{i+k}), \\
&V_k^{(i)}(x_{i+k}) = C_k^{(i)}, \\
&V_j^{(i)}(x_{i+k}) = C_j^{(i)} + (x_{i+k}-x_{i+j})V_{j+1}^{(i)}(x_{i+k}).
\end{aligned}
$$

由此可见，

$$
C_0^{(i)} = V_0^{(i)}(x_i) = f(x_i),
$$

而当 $k=1, 2, \cdots, m$ 时

$$\begin{cases} V_0^{(i)}(x_{i+k}) = f(x_{i+k}), \\ V_{j+1}^{(i)}(x_{i+k}) = \dfrac{V_j^{(i)}(x_{i+k}) - C_j^{(i)}}{x_{i+k} - x_{i+j}} \quad (j=0, 1, \cdots, k-1), \\ C_k^{(i)} = V_k^{(i)}(x_{i+k}). \end{cases} \tag{2.8}$$

根据这公式可算出 $C_0^{(i)}, C_1^{(i)}, \cdots, C_m^{(i)}$，然后按(2.7)倒算，便可算出 $P_m^{(i)}(x)$ 的值．从(2.8)可知，$C_k^{(i)}$ 是由一系列差值之商构成的，并且只同 $x_i, x_{i+1}, \cdots, x_{i+k}$ 处函数值有关，因此称为差商，记为 $f(x_i, x_{i+1}, \cdots, x_{i+k})$，并称 Newton 公式(2.6)为差商插值多项式．这种形式的插值多项式在增加一个节点 x_{i+m+1} 时，系数 $C_0^{(i)}, C_1^{(i)}, \cdots, C_m^{(i)}$ 不变，只需再算一个 $C_{m+1}^{(i)}$，因而便于增加节点，提高插值多项式的系数．不过，要是改变初始节点 x_i，系数 $C_0^{(i)}, C_1^{(i)}, \cdots, C_m^{(i)}$ 都得改变，按(2.8)计算系数就不大方便了．

差商 $f(x_i, x_{i+1}, \cdots, x_{i+k}) = C_k^{(i)}$ 同 $k+1$ 个点有关，称为 k 阶差商．它也可以通过低阶差商 $C_{k-1}^{(i+1)}$ 和 $C_{k-1}^{(i)}$ 来直接算出．事实上，比较(2.5)和(2.6)中最高次幂 x^m 的系数，可知

$$f(x_i, x_{i+1}, \cdots, x_{i+m}) = C_m^{(i)} = \sum_{k=i}^{i+m} \frac{f(x_k)}{\prod\limits_{j=i}^{i+m}{}'(x_k - x_j)}.$$

这个式子表明，差商 $f(x_i, x_{i+1}, \cdots, x_{i+m})$ 是所含节点 $x_i, x_{i+1}, \cdots, x_{i+m}$ 的函数值之线性组合，关于所含节点对称，因而同节点顺序无关．这样，从所含节点中去掉 x_i 或 x_{i+m}，分别所得低阶差商 $f(x_{i+1}, x_{i+2}, \cdots, x_{i+m}) = C_{m-1}^{(i+1)}$ 与 $f(x_i, x_{i+1}, \cdots, x_{i+m-1}) = C_{m-1}^{(i)}$ 当然也是所含节点函数值的线性组合，因此应有

$$C_m^{(i)} = \alpha C_{m-1}^{(i+1)} + \beta C_{m-1}^{(i)},$$

即

$$\sum_{k=i}^{i+m} \frac{f(x_k)}{\prod\limits_{j=i}^{i+m}{}'(x_k-x_j)} = \alpha \sum_{k=i+1}^{i+m} \frac{f(x_k)}{\prod\limits_{j=i+1}^{i+m}{}'(x_k-x_j)} + \beta \sum_{k=i}^{i+m-1} \frac{f(x_k)}{\prod\limits_{j=i}^{i+m-1}{}'(x_k-x_j)}.$$

比较两边 $f(x_i)$ 和 $f(x_{i+m})$ 的系数，可得

$$\beta = 1/(x_i - x_{i+m}), \quad \alpha = 1/(x_{i+m} - x_i) = -\beta.$$

故有

$$C_m^{(i)} = \frac{C_{m-1}^{(i+1)} - C_{m-1}^{(i)}}{x_{i+m} - x_i} = \frac{f(x_{i+1}, \cdots, x_{i+m}) - f(x_i, \cdots, x_{i+m-1})}{x_{i+m} - x_i}, \quad (2.9)$$

这便是通过低阶差商计算差商的递推公式。差商 $C_m^{(i)}$ 也可像表 1 那样列成表，只是把 $T_m^{(i)}$ 改为 $C_m^{(i)}$，并按(2.9)计算。利用这个表，在增加节点或改变起始节点时，按(2.7)计算差商插值多项式 $P_m^{(i)}(x)$ 的值，就比较方便了。

再一种，是 Neville 插值公式

$$P_0^{(i)}(x) = f(x_i),$$

$$P_m^{(i)}(x) = \frac{(x - x_i) P_{m-1}^{(i+1)}(x) - (x - x_{i+m}) P_{m-1}^{(i)}(x)}{x_{i+m} - x_i}. \quad (2.10)$$

当 $m=1$ 时，这公式变为

$$P_1^{(i)}(x) = \frac{(x - x_i) f(x_{i+1}) - (x - x_{i+1}) f(x_i)}{x_{i+1} - x_i},$$

这正是线性插值公式。$P_m^{(i)}(x)$ 可看成是由低次插值多项式按线性插值公式构造出来的。因此 Neville 插值公式也称为逐次线性插值公式。显然，$P_m^{(i)}(x)$ 是次数至多为 m 的多项式。为了证明它确实是插值多项式，尚需证明它满足插值条件(2.1)。这可用数学归纳法证明。显然 $m=0$ 时 $P_0^{(i)}(x)$ 满足

插值条件(2.1). 现设 $P_{m-1}^{(i+1)}(x)$ 和 $P_{m-1}^{(i)}(x)$ 满足插值条件(2.1), 即设

$$P_{m-1}^{(i+1)}(x_k)=f(x_k),\quad k=i+1,\ i+2,\ \cdots,\ i+m,$$
$$P_{m-1}^{(i)}(x_k)=f(x_k),\quad k=i,\ i+1,\ \cdots,\ i+m-1.$$

那么, 由(2.10)可知

$$P_m^{(i)}(x_i)=\frac{0-(x_i-x_{i+m})f(x_i)}{x_{i+m}-x_i}=f(x_i),$$

$$P_m^{(i)}(x_{i+m})=\frac{(x_{i+m}-x_i)f(x_{i+m})-0}{x_{i+m}-x_i}=f(x_{i+m}),$$

$$P_m^{(i)}(x_k)=\frac{(x_k-x_i)f(x_k)-(x_k-x_{i+m})f(x_k)}{x_{i+m}-x_i}=f(x_k)$$

$$(k=i+1,\ i+2,\ \cdots,\ i+m-1).$$

这表明 $P_m^{(i)}(x)$确实满足插值条件(2.1). $P_m^{(i)}(x)$可以排成表1那样的表格, 只要将 $T_m^{(i)}$ 改为 $P_m^{(i)}(x)$即可. 从这样的表格可见, 由 $x_i,\ x_{i+1},\ \cdots,\ x_{i+m}$ 增加一节点 x_{i+m+1}, 要算相应插值多项式 $P_{m+1}^{(i)}(x)$的值, 只要按(2.10)再算一行元素

$$P_0^{(i+m+1)}(x),\ P_1^{(i+m)}(x),\ P_2^{(i+m-1)}(x),\ \cdots,\ P_{m+1}^{(i)}(x).$$

这一行倒数第二个元素是将起始节点 x_i 改为 x_{i+1} 时的插值多项式值 $P_m^{(i+1)}(x)$. 由此可见, 逐次线性插值公式(2.10)特别适宜于变动节点、提高次数时计算插值多项式的值.

利用微分学中值定理可以证明, 插值多项式 $P_m^{(i)}(x)$的误差

$$P_m^{(i)}(x)-f(x)=\frac{f^{(m+1)}(\xi)}{(m+1)!}\prod_{k=i}^{i+m}(x-x_k),\qquad (2.11)$$

其中 ξ 在 $x_i,\ x_{i+1},\ \cdots,\ x_{i+m}$ 和 x 之间.

§2 多项式外推法及其推广

前面提到, Romberg 外推法算出的 $T_m^{(i)}$, 是 $T(h)$的插值

多项式当 $h=0$ 时的值. 现在予以证明: 令 $x=h^2$, 则据 (2.10), $f(x)=T(h)$ 满足插值条件

$$P_m^{(i)}(x_k)=f(x_k)=T(h_k)\quad (k=i,\ i+1,\ \cdots,\ i+m)$$

的插值多项式 $P_m^{(i)}(x)$, 它在 $h=0$ 时的值 $P_m^{(i)}(0)$ 应按下式计算

$$\begin{cases} P_0^{(i)}(0)=T(h_i), \\ P_m^{(i)}(0)=\dfrac{h_{i+m}^2 P_{m-1}^{(i)}(0)-h_i^2 P_{m-1}^{(i+1)}(0)}{h_{i+m}^2-h_i^2}, \end{cases}$$

比较公式(1.2)、(1.3), 便知 $T_m^{(i)}=P_m^{(i)}(0)$.

利用插值多项式的误差公式(2.11), 很容易导出 Romberg 外推算法的误差估计式. 事实上, 当 $T(h)$ 有(1.9)形式的渐近展开式时,

$$f(x)=T(h)=\tau_0+\tau_1 x+\tau_2 x^2+\tau_3 x^3+\cdots,$$

$$f^{(m+1)}(0)=(m+1)!\,\tau_{m+1},$$

故按公式(2.11)

$$\begin{aligned} T_m^{(i)}-\tau_0 &= P_m^{(i)}(0)-T(0) \\ &\approx -\tau_{m+1}\prod_{k=i}^{i+m}(-h_k^2) \\ &= (-1)^m \tau_{m+1} h_i^2 h_{i+1}^2 \cdots h_{i+m}^2. \end{aligned}$$

这一误差估计式跟(1.8)是一致的.

类似地, 如果离散化近似值 $T(h)$ 具有渐近展开式

$$T(h)=\tau_0+\tau_1 h^r+\tau_2 h^{2r}+\cdots+\tau_N h^{Nr}+\tau_{N+1}(h)h^{(N+1)r}, \tag{2.12}$$

令 $x=h^r$, 用 $f(x)=T(h)$ 的插值多项式 $P_m^{(i)}(x)$ 当 $x=0$ 时的值 $P_m^{(i)}(0)$ 去近似 $T(0)$, 即令 $T_m^{(i)}=P_m^{(i)}(0)$ 近似 τ_0, 则按公式(2.10)有

$$T_0^{(i)}=T(h_i), \tag{2.13}$$

$$\begin{cases} T_m^{(i)} = \dfrac{h_{i+m}^r T_{m-1}^{(i)} - h_i^r T_{m-1}^{(i+1)}}{h_{i+m}^r - h_i^r} \\ \quad = T_{m-1}^{(i+1)} + \dfrac{T_{m-1}^{(i+1)} - T_{m-1}^{(i)}}{(h_i/h_{i+m})^r - 1}, \end{cases} \tag{2.14}$$

而按公式(2.11)有

$$T_m^{(i)} - \tau_0 \approx (-1)^m \tau_{m+1} h_i^r h_{i+1}^r \cdots h_{i+m}^r. \tag{2.15}$$

按公式(2.13)、(2.14)推算 τ_0 近似值 $T_m^{(i)}$ 的方法，称为**多项式外推法**或 **Neville 外推算法**. 它是 1964 年 R. Bulirsch 提出来的.

Romberg 外推算法实质上是多项式外推法当 $r=2$ 时的特殊情形. 公式(2.13)与公式(1.2)完全一样，递推公式(2.14)则与公式(1.3)不同. 其它外推算法也是这样，均有 $T_0^{(i)}=T(h_i)$，只是递推公式不同. 因此，今后如无特别声明，对一种外推算法，只写出其递推公式.

如果 $T(h)$ 具有如下形式的渐近展开式

$$T(h) = \tau_0 + \tau_1 h^{r_1} + \tau_2 h^{r_2} + \cdots + \tau_N h^{r_N} + \tau_{N+1}(h) h^{r_{N+1}}, \tag{2.16}$$

其中 $r_1 < r_2 < r_3 < \cdots < r_N < r_{N+1}$，我们也可用递推公式(2.14)来计算 $T(0)$ 的较精确近似值 $T_1^{(i)}$：

$$T_1^{(i)} = T(h_{i+1}) + \frac{T(h_{i+1}) - T(h_i)}{(h_i/h_{i+1})^{r_1} - 1}, \tag{2.17}$$

即作 h^r 外推. 但是，如果 $r_j \neq jr$，对于一般的参数数列 $\{h_i\}$，却不能用递推公式(2.14)来作进一步的外推. 不过，对于特殊的、然而常用的第一种参数数列 H_1，即 $h_i = h_0 b^i$，则有类似于 Neville 算法的 **Bulirsch-Stoer 外推算法**，其递推公式为

$$\begin{aligned} T_m^{(i)} &= \frac{T_{m-1}^{(i+1)} - b^{r_m} T_{m-1}^{(i)}}{1 - b^{r_m}} \\ &= T_{m-1}^{(i+1)} + \frac{T_{m-1}^{(i+1)} - T_{m-1}^{(i)}}{b^{-r_m} - 1}. \end{aligned} \tag{2.18}$$

这种算法是1964年R. Bulirsch和J. Stoer提出来的.

可以证明,按这种算法算出的$T_m^{(i)}$,相当于满足插值条件

$$P_m^{(i)}(h_k)=T(h_k),\quad k=i,\ i+1,\ \cdots,\ i+m \tag{2.19}$$

的"多项式"

$$P_m^{(i)}(h)=a_0+a_1h^{r_1}+a_2h^{r_2}+\cdots+a_mh^{r_m}$$

当$h=0$时的值,即$T_m^{(i)}=P_m^{(i)}(0)$. 事实上,若定义一线性算子Λ_m^i,具有下列性质:

$$\begin{aligned}
&\text{(a)}\qquad \Lambda_m^iT(h)=\sum_{k=i}^{i+m}C_{mk}^{(i)}T(h_k),\\
&\text{(b)}\qquad \Lambda_m^i1=1,\\
&\text{(c)}\qquad \Lambda_m^ih^{r_j}=0,\ j=1,\ 2,\ \cdots,\ m,
\end{aligned}\tag{2.20}$$

那么,

$$\begin{aligned}
\Lambda_m^iT(h)&=\sum_{k=i}^{i+m}C_{mk}^{(i)}T(h_k)=\sum_{k=i}^{i+m}C_{mk}^{(i)}P_m^{(i)}(h_k)=\Lambda_m^iP_m^{(i)}(h)\\
&=\Lambda_m^i(a_0+a_1h^{r_1}+\cdots+a_mh^{r_m})=a_0=P_m^{(i)}(0).
\end{aligned}$$

这就是说,计算$P_m^{(i)}(0)$相当于确定$\Lambda_m^iT(h)$的值. 然而, $\Lambda_m^iT(h)$的值跟组合系数$C_{mk}^{(i)}$有关. 根据(a)和(c)知

$$\begin{aligned}
0=\Lambda_m^ih^{r_j}&=\sum_{k=i}^{i+m}C_{mk}^{(i)}h_k^{r_j}=\sum_{k=i}^{i+m}C_{mk}^{(i)}(h_0b^k)^{r_j}\\
&=h_0^{r_j}b^{ir_j}\sum_{k=0}^{m}C_{m,k+i}^{(i)}(b^{r_j})^k,\quad j=1,\ 2,\ \cdots,\ m.
\end{aligned}$$

这说明, $b^{r_j}(j=1,\ 2,\ \cdots,\ m)$恰是多项式

$$Q_m^{(i)}(x)=\sum_{k=0}^{m}C_{m,k+i}^{(i)}x^k$$

的根,因而

$$Q_m^{(i)}(x)=A\prod_{j=1}^{m}(x-b^{r_j}),$$

A为常数. 但由性质(b),

$$1=\Lambda_m^i1=\sum_{k=0}^{m}C_{m,i+k}^{(i)}\cdot1=Q_m^{(1)}(1)=A\prod_{j=1}^{m}(1-b^{r_j}),$$

所以
$$A=\frac{1}{\prod_{j=1}^{m}(1-b^{r_j})},$$
$$Q_m^{(i)}(x)=\prod_{j=1}^{m}\frac{x-b^{r_j}}{1-b^{r_j}}.$$
由此式可见，$Q_m^{(i)}(x)$跟 i 无关，从而
$$Q_m^{(i)}(x)=Q_m^{(0)}(x)=\prod_{j=1}^{m}\frac{x-b^{r_j}}{1-b^{r_j}},\tag{2.21}$$
$$C_{m,i+k}^{(i)}=C_{mk}^{(0)},$$
$$\begin{aligned}P_m^{(i)}(0)&=A_m^iP_m^{(i)}(h)=\sum_{k=i}^{i+m}C_{mk}^{(i)}T(h_k)\\&=\sum_{k=0}^{m}C_{m,i+k}^{(i)}T(h_{i+k})=\sum_{k=0}^{m}C_{mk}^{(0)}T(h_{i+k}).\end{aligned}\tag{2.22}$$
由(2.21)可得
$$Q_m^{(0)}(x)=\frac{x-b^{r_m}}{1-b^{r_m}}Q_{m-1}^{(0)}(x).$$
比较两边 x 同次幂的系数，知
$$C_{m,m}^{(0)}=\frac{C_{m-1,m-1}^{(0)}}{1-b^{r_m}},\qquad C_{m0}^{(0)}=-\frac{b^{r_m}}{1-b^{r_m}}C_{m-1,0}^{(0)},$$
$$C_{mk}^{(0)}=\frac{C_{m-1,k-1}^{(0)}-b^{r_m}C_{m-1,k}^{(0)}}{1-b^{r_m}}\quad(k=1,\ 2,\ \cdots,\ m-1).$$
于是，由(2.22)得
$$\begin{aligned}P_m^{(i)}(0)&=\frac{1}{1-b^{r_m}}\Big\{C_{m-1,m-1}^{(0)}T(h_{i+m})-b^{r_m}C_{m-1,0}^{(0)}T(h_i)\\&\quad+\sum_{k=1}^{m-1}(C_{m-1,k-1}^{(0)}-b^{r_m}C_{m-1,k}^{(0)})T(h_{i+k})\Big\}\\&=\frac{1}{1-b^{r_m}}\Big\{\sum_{k=1}^{m}C_{m-1,k-1}^{(0)}T(h_{i+k})-b^{r_m}\\&\quad\times\sum_{k=0}^{m-1}C_{m-1,k}^{(0)}T(h_{i+k})\Big\}\end{aligned}$$

$$=\frac{1}{1-b^{rm}}\left\{\sum_{k=0}^{m-1}C_{m-1,k}^{(0)}T(h_{i+1+k})-b^{rm}P_{m-1}^{(i)}(0)\right\}$$

$$=\frac{1}{1-b^{rm}}\{P_{m-1}^{(i+1)}(0)-b^{rm}P_{m-1}^{(i)}(0)\}.$$

比较(2.18)，可见计算 $T_m^{(i)}$ 与 $P_m^{(i)}(0)$ 的公式完全相同，因此 $T_m^{(i)}=P_m^{(i)}(0)$．证毕．

前面指出，多项式外推法的实质，是用 $T(h)$ 的插值多项式

$$P_m^{(i)}(h)=a_0+a_1h^r+a_2h^{2r}+\cdots+a_mh^{mr} \qquad (2.23)$$

的值 $P_m^{(i)}(0)$ 来近似极限值 $T(0)$．计算插值多项式值的方法，有 Neville 插值公式、Newton 差商插值公式和 Lagrange 插值公式．利用 Neville 插值公式，我们已得到了 Neville 外推算法(2.14)．利用后二种插值公式，则得出另外两类多项式外推算法．

利用 Newton 差商插值公式(2.9)、(2.7)时，需先算

$$C_m^{(i)}=\frac{C_{m-1}^{(i+1)}-C_{m-1}^{(i)}}{h_{i+1}^r-h_i^r},$$

然后再算

$$\begin{cases}V_m^{(i)}(0)=C_m^{(i)},\\ V_k^{(i)}(0)=C_k^{(i)}-h_{i+k}^rV_{k+1}^{(i)}(0) \quad (k=m-1,\ m-2,\ \cdots,\ 0),\\ P_m^{(i)}(0)=V_0^{(i)}(0).\end{cases}$$

比起 Neville 外推算法(2.14)来，这种算法使用起来不够方便．

利用 Lagrange 插值公式(2.5)时，可得外推公式

$$T(0)\approx P_m^{(0)}(0)=\sum_{k=0}^{m}C_{mk}^{(0)}T(h_k), \qquad (2.24)$$

其中

$$C_{mk}^{(0)}=\prod_{j=0}^{m}{}'\frac{h_j^r}{h_j^r-h_k^r}.$$

对于一些具体的参数数列 $\{h_i\}$，外推公式(2.24)可具体地写出来．

$r=1$ 时，有

$$T(0)\approx -T(h_0)+2T(h_0/2), \tag{2.25}$$

$$T(0)\approx\frac{1}{3}\{T(h_0)-6T(h_0/2)+8T(h_0/4)\},$$

$$T(0)\approx\frac{1}{2}\{T(h_0)-8T(h_0/2)+9T(h_0/3)\},$$

$$T(0)\approx\frac{1}{6}\{-T(h_0)+24T(h_0/2)-81T(h_0/3)$$
$$+64T(h_0/4)\},$$

$$T(0)\approx\frac{1}{21}\{-T(h_0)+14T(h_0/2)-56T(h_0/4)$$
$$+64T(h_0/4)\},$$

$$T(0)\approx\frac{1}{6}\{-8T(h_0/2)+81T(h_0/3)-192T(h_0/4)$$
$$+125T(h_0/5)\},$$

$$T(0)\approx\frac{1}{6}\{-343T(h_0/7)+1536T(h_0/8)$$
$$-2187T(h_0/9)+1000T(h_0/10)\},$$

$$T(0)\approx\frac{1}{2}\{-576T(h_0/12)+2197T(h_0/13)$$
$$-2744T(h_0/14)+1125T(h_0/15)\},$$

$$T(0)\approx\frac{1}{6}\{-4913T(h_0/17)+17496T(h_0/18)$$
$$-20577T(h_0/19)+8000T(h_0/20)\},$$

$$T(0)\approx\frac{1}{24}\{T(h_0)-64T(h_0/2)+486T(h_0/3)$$
$$-1024T(h_0/4)+625T(h_0/5)\},$$

$$T(0)\approx\frac{1}{315}\{T(h_0)-30T(h_0/2)+280T(h_0/4)$$
$$-960T(h_0/8)+1024T(h_0/16)\},$$

$$T(0)\approx\frac{1}{30}\{T(h_0)-60T(h_0/2)+405T(h_0/3)$$
$$-640T(h_0/4)+324T(h_0/6)\}.$$

$r=2$ 时，有

$$T(0)\approx\frac{1}{3}\{-T(h_0)+4T(h_0/2)\},\tag{2.26}$$

$$T(0)\approx\frac{1}{120}\{5T(h_0)-128T(h_0/2)+243T(h_0/3)\},$$

$$T(0)\approx\frac{1}{45}\{T(h_0)-20T(h_0/2)+64T(h_0/4)\},$$

$$T(0)\approx\frac{1}{2520}\{-7T(h_0)+896T(h_0/2)$$
$$-6561T(h_0/3)+8192T(h_0/4)\},$$

$$T(0)\approx\frac{1}{2835}\{-T(h_0)+84T(h_0/2)-1344T(h_0/4)$$
$$+4096T(h_0/8)\},$$

$$T(0)\approx\frac{1}{362880}\{42T(h_0)-24576T(h_0/2)$$
$$+531441T(h_0/3)-2097152T(h_0/4)$$
$$+1953125T(h_0/5)\},$$

$$T(0)\approx\frac{1}{12600}\{T(h_0)-560T(h_0/2)+10935T(h_0)$$
$$-32768T(h_0/4)+34992T(h_0/6)\},$$

$$T(0)\approx\frac{1}{722925}\{T(h_0)-340T(h_0/2)+22848T(h_0/4)$$
$$-348160T(h_0/8)+1048576T(h_0/16)\}.$$

利用(2.25)或(2.26)中某一公式，或由(2.24)导出的其它公式，来推算 $T(0)$ 的算法，统称为 **Lagrange 外推算法**. (2.25)中第六到第九个公式，是 1954 年 E. Salzer 提出来的.

显然，对于固定的参数$\{h_i\}$，利用 Lagrange 外推算法是十分方便的．但要改变参数 h_i，就不方便了．

§3 广义多项式外推法

在一般情况下，离散化近似值 $T(h)$与极限值 $T(0)$之间，可能存在如下形式的渐近展开式

$$T(h)=\tau_0\psi_0(h)+\tau_1\psi_1(h)+\cdots+\tau_N\psi_N(h)+\tau_{N+1}(h)\psi_{N+1}(h), \tag{2.27}$$

其中 $\tau_0, \tau_1, \cdots, \tau_N$ 为常数，$\tau_{N+1}(h)$有界，

$$\psi_0(h)=1, \quad \lim_{h\to 0}\psi_{j+1}(h)/\psi_j(h)=0. \tag{2.28}$$

此时，我们自然想用满足插值条件(2.19)的"多项式"

$$P_m^{(i)}(h)=a_0\psi_0(h)+a_1\psi_1(h)+\cdots+a_m\psi_m(h) \tag{2.29}$$

当 $h=0$ 时的值 $T_m^{(i)}=P_m^{(i)}(0)$来推算 $T(0)=\tau_0$.

为使插值"多项式" $P_m^{(i)}(h)$存在而且唯一，需要插值条件(2.19)对应的方程组

$$a_0\psi_0(h_k)+a_1\psi_1(h_k)+\cdots+a_m\psi_m(h_k)=T(h_k) \quad (k=i,\ i+1,\ \cdots,\ i+m) \tag{2.30}$$

的解$(a_0, a_1, \cdots, a_m)$存在而且唯一．由线性方程组的理论知道，这相当于要求方程组(2.30)的系数行列式不等于 0，即广义 Vandermonde 行列式

$$V\begin{pmatrix}\psi_0, \cdots, \psi_m\\ h_i, \cdots, h_{i+m}\end{pmatrix}=\begin{vmatrix}\psi_0(h_i) & \psi_1(h_i) & \cdots & \psi_m(h_i)\\ \psi_0(h_{i+1}) & \psi_1(h_{i+1}) & \cdots & \psi_m(h_{i+1})\\ \cdots & \cdots & \cdots & \cdots\\ \psi_0(h_{i+m}) & \psi_1(h_{i+m}) & \cdots & \psi_m(h_{i+m})\end{vmatrix}\neq 0. \tag{2.31}$$

对区间 I 上任意 $m+1$ 个点 $h_i, h_{i+1}, \cdots, h_{i+m}$ 满足(2.31)的连续函数组$(\psi_0, \psi_1, \cdots, \psi_m)$称为**切比雪夫函数组**. 因此, 对区间 I 上任意 $m+1$ 个点 $h_i, h_{i+1}, \cdots, h_{i+m}$, 要求满足插值条件(2.30)的"多项式" $P_m^{(i)}(h)$存在而且唯一, 相当于要求$(\psi_0, \psi_1, \cdots, \psi_m)$是切比雪夫函数组. 下面, 我们假定$(\psi_0, \psi_1, \cdots, \psi_m)$是包含 $h=0$ 的某个区间 I 上的切比雪夫函数组, 并称"多项式"(2.29)为**广义插值多项式**.

由于条件(2.31)也是线性齐次方程组

$$a_0\psi_0(h_k)+a_1\psi_1(h_k)+\cdots+a_m\psi_m(h_k)=0$$
$$(k=i, i+1, \cdots, i+m) \qquad (2.32)$$

没有非零解的充要条件, 因此说$(\psi_0, \psi_1, \cdots, \psi_m)$是区间 I 上的切比雪夫函数组, 也等价于说, $\psi_0, \psi_1, \cdots, \psi_m$ 的任何非零的线性组合, 即系数 $a_0, a_1, \cdots, a_m$ 不全为零的广义多项式, 在区间 I 上至多有 m 个零点. 设有函数序列 $\{\psi_i\}$, 如果对任意 m 来说$(\psi_0, \psi_1, \cdots, \psi_m)$都是区间 I 上的切比雪夫函数组, 则称 $\{\psi_i\}$ 为区间 I 上的**完全切比雪夫函数系**. 以后我们假定 $\{\psi_i\}$ 是在包含 $h=0$ 的某个区间 I 上的完全切比雪夫函数系. 利用(2.4)不难验证, $\{h^{i}\}$是任意区间 I 上的完全切比雪夫函数系.

现在我们来导出计算广义插值多项式 $P_m^{(i)}(h)$的递推公式. 由于从(2.30)解出的 $a_0, a_1, \cdots, a_m$ 是 $T(h_k)$ $(k=i, i+1, \cdots, i+m)$的线性组合, 代入(2.29), $P_m^{(i)}(h)$当然也是$T(h_k)$的线性组合, 组合系数同 $T(h)$无关. $P_{m-1}^{(i+1)}(h)$和 $P_{m-1}^{(i)}(h)$的情况也是如此. 所以, 可设想

$$P_m^{(i)}(h)=\lambda(h)P_{m-1}^{(i+1)}(h)+\mu(h)P_{m-1}^{(i)}(h). \qquad (2.33)$$

由广义插值多项式的唯一性可知, $T(h)=\psi_0(h)=1$ 时,

$$P_m^{(i)}(h)=P_{m-1}^{(i+1)}(h)=P_{m-1}^{(i)}(h)=\psi_0(h)=1,$$

(2.33)变为

$$1=\lambda(h)+\mu(h), \quad \text{所以} \quad \mu(h)=1-\lambda(h).$$

当 $T(h)=\psi_m(h)$ 时，$P_m^{(i)}(h)=\psi_m(h)$，分别记这时的广义插值多项式 $P_{m-1}^{(i+1)}(h)$、$P_{m-1}^{(i)}(h)$ 为 $\widetilde{P}_{m-1}^{(i+1)}(h)$、$\widetilde{P}_{m-1}^{(i)}(h)$，则(2.33)变为

$$\psi_m(h)=\lambda(h)\widetilde{P}_{m-1}^{(i+1)}(h)+\mu(h)\widetilde{P}_{m-1}^{(i)}(h).$$

由此可解出

$$\lambda(h)=\frac{\psi_m(h)-\widetilde{P}_{m-1}^{(i)}(h)}{\widetilde{P}_{m-1}^{(i+1)}-\widetilde{P}_{m-1}^{(i)}(h)},$$

$$\mu(h)=-\frac{\psi_m(h)-\widetilde{P}_{m-1}^{(i+1)}(h)}{\widetilde{P}_{m-1}^{(i+1)}-\widetilde{P}_{m-1}^{(i)}(h)}. \tag{2.34}$$

当 $h\neq h_i, h_{i+1}, \cdots, h_{i+m}$ 时，这里的分母不会等于 0. 这是因为，它是 $\psi_0, \psi_1, \cdots, \psi_{m-1}$ 的线性组合，$h_{i+1}, h_{i+2}, \cdots, h_{i+m-1}$ 是它的 $m-1$ 个零点，如果组合系数不全为 0，由于 $\psi_0, \psi_1, \cdots, \psi_{m-1}$ 为切比雪夫函数组，它就不应再有零点；如果组合系数全为 0，则 $\widetilde{P}_{m-1}^{(i+1)}(h)\equiv\widetilde{P}_{m-1}^{(i)}(h)$，它们的公共系数 $a_0, a_1, \cdots, a_{m-1}$ 满足对应于(2.30)的方程组

$$a_0\psi_0(h_k)+a_1\psi_1(h_k)+\cdots+a_{m-1}\psi_{m-1}(h_k)-1\cdot\psi_m(h_k)=0$$
$$(k=i, i+1, \cdots, i+m).$$

这方程组有非零解 $(a_0, a_1, \cdots, a_{m-1}, -1)$，故系数行列式

$$V\begin{pmatrix}\psi_0, \psi_1, \cdots, \psi_m \\ h_i, h_{i+1}, \cdots, h_{i+m}\end{pmatrix}=0,$$

这同(2.31)矛盾.

将(2.34)代入(2.33)，并令 $e_m^{(i)}T(h)$ 表示广义插值多项式 $P_m^{(i)}(h)$ 的误差，即

$$e_m^{(i)}T(h)=P_m^{(i)}(h)-T(h), \tag{2.35}$$

则得

$$P_m^{(i)}(h)=\frac{e_{m-1}^{(i)}\psi_m(h)P_{m-1}^{(i+1)}(h)-e_{m-1}^{(i+1)}\psi_m(h)P_{m-1}^{(i)}(h)}{e_{m-1}^{(i)}\psi_m(h)-e_{m-1}^{(i+1)}\psi_m(h)}$$

$$=P_{m-1}^{(i+1)}(h)+\frac{P_{m-1}^{(i+1)}(h)-P_{m-1}^{(i)}(h)}{\dfrac{e_{m-1}^{(i)}\psi_m(h)}{e_{m-1}^{(i+1)}\psi_m(h)}-1}. \qquad (2.36)$$

将 $T(h)=\psi_j(h)\,(j>m)$ 代入，并注意(2.35)，则得

$$e_m^{(i)}\psi_j(h)=\frac{e_{m-1}^{(i)}\psi_m(h)P_{m-1}^{(i+1)}(h)-e_{m-1}^{(i+1)}\psi_m(h)P_{m-1}^{(i)}(h)}{e_{m-1}^{(i)}\psi_m(h)-e_{m-1}^{(i+1)}\psi_m(h)}-\psi_j(h)$$

$$=\{e_{m-1}^{(i)}\psi_m(h)[P_{m-1}^{(i+1)}(h)-\psi_j(h)]$$
$$-e_{m-1}^{(i+1)}\psi_m(h)[P_{m-1}^{(i)}(h)-\psi_j(h)]\}$$
$$/[e_{m-1}^{(i)}\psi_m(h)-e_{m-1}^{(i+1)}\psi_m(h)],$$

即

$$e_m^{(i)}\psi_j(h)=\frac{e_{m-1}^{(i)}\psi_m(h)e_{m-1}^{(i+1)}\psi_j(h)-e_{m-1}^{(i+1)}\psi_m(h)e_{m-1}^{(i)}\psi_j(h)}{e_{m-1}^{(i)}\psi_m(h)-e_{m-1}^{(i+1)}\psi_m(h)}$$

$$=e_{m-1}^{(i+1)}\psi_j(h)+\frac{e_{m-1}^{(i+1)}\psi_j(h)-e_{m-1}^{(i+1)}\psi_j(h)}{\dfrac{e_{m-1}^{(i)}\psi_m(h)}{e_{m-1}^{(i+1)}\psi_m(h)}-1}. \qquad (2.37)$$

公式(2.36)就是计算广义插值多项式 $P_m^{(i)}(h)$ 的递推公式，其中用到的 $e_{m-1}^{(i)}\psi_m(h)$ 和 $e_{m-1}^{(i+1)}\psi_m(h)$ 可根据递推公式(2.37)预先算出．当然，$m=0$ 时

$$P_0^{(i)}(h)=T(h_i),$$
$$e_0^{(i)}\psi_j(h)=\psi_j(h_i)-\psi_j(h).$$

为了利用这些公式推算极限值 $T(0)$，令 $h=0$，仍记 $P_m^{(i)}(0)=T_m^{(i)}$，注意由条件(2.28)知 $\psi_j(0)=0\,(j>0)$，便得

$$\begin{cases} T_0^{(i)}=T(h_i), \\ T_m^{(i)}=\dfrac{e_{m-1}^{(i)}\psi_m(0)T_{m-1}^{(i+1)}-e_{m-1}^{(i+1)}\psi_m(0)T_{m-1}^{(i)}}{e_{m-1}^{(i)}\psi_m(0)-e_{m-1}^{(i+1)}\psi_m(0)} \\ \qquad =T_{m-1}^{(i+1)}+\dfrac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\dfrac{e_{m-1}^{(i)}\psi_m(0)}{e_{m-1}^{(i+1)}\psi_m(0)}-1}, \end{cases} \qquad (2.38)$$

$$\begin{cases} e_0^{(i)}\psi_j(0)=\psi_j(h_i), \\ e_m^{(i)}\psi_j(0)=\dfrac{e_{m-1}^{(i)}\psi_m(0)e_{m-1}^{(i+1)}\psi_j(0)-e_{m-1}^{(i+1)}\psi_m(0)e_{m-1}^{(i)}\psi_j(0)}{e_{m-1}^{(i)}\psi_m(0)-e_{m-1}^{(i+1)}\psi_m(0)} \\ \qquad =e_{m-1}^{(i+1)}\psi_j(0)+\dfrac{e_{m-1}^{(i+1)}\psi_j(0)-e_{m-1}^{(i)}\psi_j(0)}{\dfrac{e_{m-1}^{(i)}\psi_m(0)}{e_{m-1}^{(i+1)}\psi_m(0)}-1}. \end{cases} \tag{2.39}$$

利用递推公式(2.38)推算$T(0)$近似值$T_m^{(i)}$的方法称为 **Mühlbach 外推算法**. 它是 1976 年 G. Mühlbach 提出来的. (2.38)中所用的$e_{m-1}^{(i)}\psi_m(0)$和$e_{m-1}^{(i+1)}\psi_m(0)$可根据(2.39)算出.

显然, 递推公式(2.38)类似于(1.3)、(2.14)和(2.18), 但比较复杂, 需要另外计算$e_{m-1}^{(i)}\psi_m(0)$和$e_{m-1}^{(i+1)}\psi_m(0)$. 不过, 这些量同$T(h)$无关, 一旦选定$\{\psi_i\}$和$\{h_i\}$, 就可预先算出. 注意(2.39)类似于(2.38), 只要把(2.38)中的$T_{m-1}^{(i)}$和$T_{m-1}^{(i+1)}$分别换为$e_{m-1}^{(i)}\psi_j(0)$和$e_{m-1}^{(i+1)}\psi_j(0)$, 就变为(2.39), 因而$e_m^{(i)}\psi_j(0)$和$T_m^{(i)}$的计算过程相同. 然而, 根据广义插值多项式的唯一性可知, 当$m\geqslant j$时$e_m^{(i)}\psi_j(h)=0$, 所以实际只需计算$e_0^{(i)}\psi_j(0)$, $e_1^{(i)}\psi_j(0)$, $\cdots$, $e_{j-1}^{(i)}\psi_j(0)$.

余项$e_m^{(i)}T(h)$和$e_m^{(i)}\psi_j(0)$还可用广义 Vandermonde 行列式来表示和计算. 事实上, 由插值条件(2.30)和$e_m^{(i)}T(h)$的定义(2.35)有

$$\begin{cases} a_0\psi_0(h_k)+\cdots+a_m\psi_m(h_k)-1\cdot T(h_k)=0 \\ \qquad\qquad (k=i,\ i+1,\ \cdots,\ i+m), \\ a_0\psi_0(h_k)+\cdots+a_m\psi_m(h)-1\cdot[e_m^{(i)}T(h)+T(h)]=0. \end{cases}$$

这是$(a_0,\ a_1,\ \cdots,\ a_m,\ -1)$的线性齐次方程组, 它有非零解, 故

$$\begin{vmatrix} \psi_0(h_i) & \cdots & \psi_m(h_i) & T(h_i)+0 \\ \cdots & \cdots & \cdots & \cdots \\ \psi_0(h_{i+m}) & \cdots & \psi_m(h_{i+m}) & T(h_{i+m})+0 \\ \psi_0(h) & \cdots & \psi_m(h) & T(h)+e_m^{(i)}T(h) \end{vmatrix}=0,$$

从而

$$e_m^{(i)}T(h) = -V\begin{pmatrix}\psi_0, & \cdots, & \psi_m, & T\\ h_i, & \cdots, & h_{i+m}, & h\end{pmatrix}\Big/ V\begin{pmatrix}\psi_0, & \cdots, & \psi_m\\ h_i, & \cdots, & h_{i+m}\end{pmatrix}. \tag{2.40}$$

令 $T(h)=\psi_j(h)$, $h=0$, 得

$$e_m^{(i)}\psi_j(0) = -V\begin{pmatrix}\psi_0, & \cdots, & \psi_m, & \psi_j\\ h_i, & \cdots, & h_{i+m}, & 0\end{pmatrix}\Big/ V\begin{pmatrix}\psi_0, & \cdots, & \psi_m\\ h_i, & \cdots, & h_{i+m}\end{pmatrix}.$$

注意 $\psi_0(h)=1$, $\psi_j(0)=0(j>0)$, 上式中分子

$$\begin{vmatrix}\psi_0(h_i) & \cdots & \psi_m(h_i) & \psi_j(h_i)\\ \cdots & \cdots & \cdots & \cdots\\ \psi_0(h_{i+m}) & \cdots & \psi_m(h_{i+m}) & \psi_j(h_{i+m})\\ 1 & \cdots & 0 & 0\end{vmatrix}$$

$$=(-1)^{m+3}V\begin{pmatrix}\psi_1, & \cdots, & \psi_m, & \psi_j\\ h_i, & \cdots\cdots, & & h_{i+m}\end{pmatrix},$$

故得

$$e_m^{(i)}\psi_j(0) = (-1)^m V\frac{\begin{pmatrix}\psi_1, & \cdots, & \psi_m, & \psi_j\\ h_i, & \cdots\cdots, & & h_{i+m}\end{pmatrix}}{V\begin{pmatrix}\psi_0, & \cdots, & \psi_m\\ h_i, & \cdots\cdots, & h_{i+m}\end{pmatrix}}. \tag{2.41}$$

在 $\{\psi_j\}=\{h^{jr}\}$ 的具体情况下，从(2.41)分子行列式的第一行中提出公因子 h_i^r, 从第二行中提出公因子 h_{i+1}^r, $\cdots$, 从第 $m+1$ 行中提出公因子 h_{i+m}^r, 则得

$$e_m^{(i)}\psi_j(0) = (-1)^m h_i^r h_{i+1}^r \cdots h_{i+m}^r \frac{V\begin{pmatrix}\psi_0, & \cdots, & \psi_{m-1}, & \psi_{j-1}\\ h_i, & \cdots\cdots, & & h_{i+m}\end{pmatrix}}{V\begin{pmatrix}\psi_0, & \cdots, & \psi_m\\ h_i, & \cdots, & h_{i+m}\end{pmatrix}}.$$

由此可见

$$\begin{cases} e_m^{(i)}\psi_{m+1}(0) = (-1)^m h_i^r h_{i+1}^r \cdots h_{i+m}^r,\\ e_m^{(i)}\psi_j(0) = (-1)^m h_i^r h_{i+1}^r \cdots h_{i+m}^r \times o(h_i^r) \quad (j>m+1),\end{cases} \tag{2.42}$$

其中 $o(h_i^r)$ 表示 h_i^r 的高阶无穷小量. 于是

$$e_{m-1}^{(i)}\psi_m(0)/e_{m-1}^{(i+1)}\psi_m(0)=h_i^r/h_{i+m}^r.$$

代入(2.38), 则得到(2.14). 这说明, Neville 外推算法(2.14)及 $r=2$ 时的 Romberg 外推算法(1.3), 是 Mühlbach 外推算法的特殊情形.

在 $\{\psi_j\}=\{h^{r_j}\}$ 而 $h_k=h_0b^k$ 的特殊情况下,

$$\frac{e_{m-1}^{(i)}\psi_m(0)}{e_{m-1}^{(i+1)}\psi_m(0)}=\frac{V\begin{pmatrix}\psi_1, & \cdots, & \psi_m\\ h_i, & \cdots, & h_{i+m-1}\end{pmatrix}}{V\begin{pmatrix}\psi_1, & \cdots, & \psi_m\\ h_{i+1}, & \cdots, & h_{i+m}\end{pmatrix}}\cdot\frac{V\begin{pmatrix}\psi_0, & \cdots, & \psi_{m-1}\\ h_{i+1}, & \cdots, & h_{i+m}\end{pmatrix}}{V\begin{pmatrix}\psi_0, & \cdots, & \psi_{m-1}\\ h_i, & \cdots, & h_{i+m-1}\end{pmatrix}}.$$

对右边第一个分式的分母行列式,由各列分别提出公因子 b^{r_1}, b^{r_2}, $\cdots$, b^{r_m}, 则所得行列式与分子相同; 对第二个分式的分子行列式, 由各列分别提出 1, b^{r_1}, $\cdots$, $b^{r_{m-1}}$, 则所得行列式与分母相同. 于是

$$\frac{e_{m-1}^{(i)}\psi_m(0)}{e_{m-1}^{(i+1)}\psi_m(0)}=\frac{1}{b^{r_1}b^{r_2}\cdots b^{r_m}}\cdot b^{r_1}\cdots b^{r_{m-1}}=b^{-r_m}.$$

将此结果代入(2.38), 得到(2.18). 这说明, Bulirsch-Stoer 外推算法是 Mühlbach 外推算法的特殊情形.

对于 $\{\psi_j(h)\}=\{h^{r_j}\}$, 按照公式(2.39)可得

$$e_0^{(i)}\psi_j(0)=\psi_j(h_i)=h_i^{r_j}=h_0b^{ir_j},$$

$$e_1^{(i)}\psi_j(0)=\frac{h_0^{r_1}b^{ir_1}\cdot h_0^{r_j}b^{(i+1)r_j}-h_0^{r_1}b^{(i+1)r_1}\cdot h_0^{r_j}b^{ir_j}}{h_0^{r_1}b^{ir_1}-h_0^{r_1}b^{(i+1)r_1}}$$

$$=h_i^{r_j}\frac{b^{r_j}-b^{r_1}}{1-b^{r_1}},$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$e_m^{(i)}\psi_j(0)=h_i^{r_j}\prod_{k=1}^{m}\frac{b^{r_j}-b^{r_k}}{1-b^{r_k}}. \tag{2.43}$$

§4 广义多项式插值法与 Richardson 外推法的关系

广义插值多项式 $P_m^{(i)}(h)$ 可以看成从 $T(h)$ 的近似多项式

$$P_0^{(i)}(h)=T(h_i)$$

中逐步消去误差项 $\tau_1\psi_1(h)$、$\tau_2\psi_2(h)$、…的结果. 证明如下:
由(2.27), 即

$$T(h)=\tau_0+\tau_1\psi_1(h)+\tau_2\psi_2(h)+\cdots,$$

我们有

$$P_0^{(i)}(h)=T(h_i)=\tau_0+\tau_1\psi_1(h_i)+\tau_2\psi_2(h_i)+\cdots.$$

二式相减, 得

$$\begin{aligned}P_0^{(i)}(h)&=T(h)+\tau_1[\psi_1(h_i)-\psi_1(h)]\\&\quad+\tau_2[\psi_2(h_i)-\psi_2(h)]+\cdots\\&=T(h)+\tau_1e_0^{(i)}\psi_1(h)+\tau_2e_0^{(i)}\psi_2(h)+\cdots,\end{aligned}\qquad(2.44)$$

这里 $\tau_1e_0^{(i)}\psi_1(h)$、$\tau_2e_0^{(i)}\psi_2(h)$、… 是 $T(h)$ 的近似多项式 $P_0^{(i)}(h)$ 包含的误差项. 由此式得

$$P_0^{(i+1)}=T(h)+\tau_1e_0^{(i+1)}\psi_1(h)+\tau_2e_0^{(i+1)}\psi_2(h)+\cdots.$$

将此式和(2.44)分别乘 $e_0^{(i)}\psi_1(h)$、$e_0^{(i+1)}\psi_1(h)$, 然后相减, 并除以 $e_0^{(i)}\psi_1(h)-e_0^{(i+1)}\psi_1(h)$, 则得

$$\frac{e_0^{(i)}\psi_1(h)P_0^{(i+1)}(h)-e_0^{(i+1)}\psi_1(h)P_0^{(i)}(h)}{e_0^{(i)}\psi_1(h)-e_0^{(i+1)}\psi_1(h)}$$

$$=T(h)+\tau_2\frac{e_0^{(i)}\psi_1(h)e_0^{(i+1)}\psi_2(h)-e_0^{(i+1)}\psi_1(h)e_0^{(i)}\psi_2(h)}{e_0^{(i)}\psi_1(h)-e_0^{(i+1)}\psi_1(h)}+\cdots,$$

根据(2.36)、(2.37), 此式即

$$P_1^{(i)}(h)=T(h)+\tau_2e_1^{(i)}\psi_2(h)+\tau_3e_1^{(i)}\psi_3(h)+\cdots.$$

这说明, $P_1^{(i)}(h)$ 是从 $P_0^{(i)}(h)=T(h_i)$ 中消去首项误差 $e_0^{(i)}\psi_1(h)$ 的结果, 它只含有 $\tau_2e_1^{(i)}\psi_2(h)$、$\tau_3e_1^{(i)}\psi_3(h)$、…等项误差. 一

般，当 $m<N$ 时，设有

$$P_{m-1}^{(i)}(h)=T(h)+\tau_m e_{m-1}^{(i)}\psi_m(h)+\tau_{m+1}e_{m-1}^{(i)}\psi_{m+1}(h)+\cdots$$

则

$$P_{m-1}^{(i+1)}(h)=T(h)+\tau_m e_{m-1}^{(i+1)}\psi_m(h)+\tau_{m+1}e_{m-1}^{(i+1)}\psi_{m+1}(h)+\cdots.$$

由此二式得

$$\frac{e_{m-1}^{(i)}\psi_m(h)P_{m-1}^{(i+1)}(h)-e_{m-1}^{(i+1)}\psi_m(h)P_{m-1}^{(i)}(h)}{e_{m-1}^{(i)}\psi_m(h)-e_{m-1}^{(i+1)}\psi_m(h)}$$
$$=T(h)+\tau_{m+1}\left[e_{m-1}^{(i)}\psi_m(h)e_{m-1}^{(i+1)}\psi_{m+1}(h)-e_{m-1}^{(i+1)}\psi_m(h)e_{m-1}^{(i)}\psi_{m+1}(h)\right]/\left[e_{m-1}^{(i)}\psi_m(h)-e_{m-1}^{(i+1)}\psi_m(h)\right]+\cdots.$$

根据(2.36)、(2.37)，此式亦即

$$P_m^{(i)}(h)=T(h)+\tau_{m+1}e_m^{(i)}\psi_{m+1}(h)+\tau_{m+2}e_m^{(i)}\psi_{m+2}(h)+\cdots. \tag{2.45}$$

这样，我们便用数学归纳法证明了(2.45)，即证明了广义插值多项式 $P_m^{(i)}(h)$ 是由 $T(h)$ 的近似式 $T(h_i)=P_0^{(i)}(h)$ 中逐步消去误差项的结果；它作为 $T(h)$ 的近似式，只含有首项是 $\tau_{m+1}e_m^{(i)}\psi_{m+1}(h)$ 的一些误差项．或者说，广义插值多项式 $P_m^{(i)}(h)$，是对 $T(h_i)$ 应用 Richardson 外推法的结果.

作为特例，$T_m^{(i)}=P_m^{(i)}(0)$ 自然也是从 $T(0)$ 的近似值 $T(h_i)$ 中逐步消去误差项的结果．这说明，广义多项式外推法，即 Mühlbach 外推法，也是 Richardson 外推法．这还说明，对固定的 i，随着 m 的增加，$T_m^{(i)}$ 的误差阶数逐步提高．正是由于这个原因，广义多项式外推法，可用来加速 $\{T(h_i)\}$ 的收敛，成为数值计算的有力工具．当然，这些话都只是在 $m<N$（见渐近展开式(2.27)）时才成立．如果 $m\geqslant N$，那就未必收敛了．渐近展开式(2.27)不存在，可看作 $N=0$ 或 $\tau_{N+1}(h)$ 不一定有界．这时广义多项式外推法未必加速收敛.

§5 误差估计

由公式(2.45)立即可得 $T_m^{(i)}$ 的误差表示式

$$T_m^{(i)}-T(0)=\tau_{m+1}e_m^{(i)}\psi_{m+1}(0)+\tau_{m+2}e_m^{(i)}\psi_{m+2}(0)+\cdots. \tag{2.46}$$

当然，这里 $m<N$.

在 $\{\psi_j\}=\{h^{jr}\}$ 的情况下，即对多项式外推法，由(2.46)和(2.42)得到

$$T_m^{(i)}-T(0)=(-1)^m h_i^r h_{i+1}^r\cdots h_{i+m}^r\{\tau_{m+1}+o(h_i^r)\}. \tag{2.47}$$

在 $\{\psi_j\}=\{h^{r_j}\}$ 而 $h_i=h_0b^i(0<b<1)$ 的情况下，即对 Bulirsch-Stoer 外推算法，由(2.46)和(2.43)得

$$\begin{aligned}T_m^{(i)}-T(0)&=\tau_{m+1}h_i^{r_{m+1}}\prod_{k=1}^{m}\frac{b^{r_{m+1}}-b^{r_k}}{1-b^{r_k}}\\&\quad+\tau_{m+2}h_i^{r_{m+2}}\prod_{k=1}^{m}\frac{b^{r_{m+2}}-b^{r_k}}{1-b^{r_k}}+\cdots\\&=(-1)^m h_i^{r_{m+1}}b^{\sum_{k=1}^{m}r_k}\prod_{k=1}^{m}\frac{1-b^{r_{m+1}-r_k}}{1-b^{r_k}}\\&\quad\times\{\tau_{m+1}+o(h_i^{r_{m+2}-r_{m+1}})\}.\end{aligned} \tag{2.48}$$

利用类似于(2.20)定义的线性外推算子，我们可以导出 $T_m^{(i)}$ 的误差的另一种表示式.

设线性外推算子 Λ_m^i 具有如下性质：

$$\begin{cases}(a)\ \Lambda_m^i T(h)=\sum\limits_{k=i}^{i+m}C_{mk}^{(i)}T(h_k),\\(b)\ \Lambda_m^i 1=1,\\(c)\ \Lambda_m^i\psi_j(h)=0\quad(j=1,\ 2,\ \cdots,\ m).\end{cases} \tag{2.49}$$

那么，对于广义插值多项式 $P_m^{(i)}(h)$,

$$\Lambda_m^i T(h) = \sum_{k=i}^{i+m} C_{mk}^{(i)} T(h_k) = \sum_{k=i}^{i+m} C_{mk}^{(i)} P_m^{(i)}(h_k) = \Lambda_m^i P_m^{(i)}(h)$$
$$= \Lambda_m^i \left(\sum_{j=0}^{m} a_j \psi_j(h) \right) = a_0 = P_m^{(i)}(0).$$

这说明，$T_m^{(i)} = P_m^{(i)}(0)$不仅可以看成是从$P_0^{(i)}(0)$中逐步消去低次误差项的结果，还可以看成是线性外推算子Λ_m^i作用于$T(h)$的结果.

(2.49)中的$C_{mk}^{(i)}$可由(2.40)求出. 事实上，将(2.40)右边分子行列式按最后一列展开，则得

$$P_m^{(i)}(h) - T(h) = e_m^{(i)} T(h)$$
$$= \sum_{k=i}^{i+m} \frac{V\begin{pmatrix} \psi_0, & \cdots\cdots\cdots\cdots\cdots\cdots, & \psi_m \\ h_i, \cdots, h_{k-1}, & h, h_{k+1}, \cdots, & h_{i+m} \end{pmatrix}}{V\begin{pmatrix} \psi_0, & \cdots, & \psi_m \\ h_i, & \cdots, & h_{i+m} \end{pmatrix}} T(h_k) - T(h),$$

可见

$$P_m^{(i)}(h) = \sum_{k=i}^{i+m} \frac{V\begin{pmatrix} \psi_0, & \cdots\cdots\cdots\cdots\cdots\cdots\cdots, & \psi_m \\ h_i, \cdots, h_{k-1}, & h, & h_{k+1}, \cdots, h_{i+m} \end{pmatrix}}{V\begin{pmatrix} \psi_0, & \cdots, & \psi_m \\ h_i, & \cdots, & h_{i+m} \end{pmatrix}} T(h_k). \tag{2.50}$$

此式用函数值$T(h_i)$、$T(h_{i+1})$、…、$T(h_{i+m})$的线性组合表出了广义插值多项式，而且当$\psi_j(h) = h^j$时变为通常的 Lagrange 插值多项式，故称为广义 Lagrange 插值多项式. 令$h=0$，(2.50)左方成为$T_m^{(i)}$，故据(2.49)(a)得

$$C_{mk}^{(i)} = \frac{V\begin{pmatrix} \psi_0, & \cdots\cdots\cdots\cdots\cdots\cdots\cdots & \psi_m \\ h_i, \cdots, h_{k-1}, & 0, & h_{k+1}, \cdots, h_{i+m} \end{pmatrix}}{V\begin{pmatrix} \psi_0, & \cdots, & \psi_m \\ h_i, & \cdots, & h_{i+m} \end{pmatrix}}. \tag{2.51}$$

特别地，当$\psi_j(h) = h^{jr}$时，由此可得

$$C_{mk}^{(i)}=\prod_{j=i}^{i+m}{}' \frac{h_j^r}{h_j^r-h_k^r}. \tag{2.52}$$

这一结果，跟在 Lagrange 插值公式(2.5)中令 $x=h^r$ 所得结果是一致的.

现在，把线性外推算子 A_m^i 作用于渐近展开式(2.27)，则当 $m\geqslant N$ 时

$$\begin{aligned}T_m^{(i)}&=A_m^i T(h)=A_m^i\left\{\sum_{j=0}^{N}\tau_j\psi_j(h)+\tau_{N+1}(h)\psi_{N+1}(h)\right\}\\&=\tau_0+A_m^i\{\tau_{N+1}(h)\psi_{N+1}(h)\}\\&=\tau_0+\sum_{k=i}^{i+m}C_{mk}^{(i)}\tau_{N+1}(h_k)\psi_{N+1}(h_k),\end{aligned}$$

故得 $T_m^{(i)}$ 的另一种误差表达式

$$T_m^{(i)}-\tau_0=\sum_{k=i}^{i+m}C_{mk}^{(i)}\tau_{N+1}(h_k)\psi_{N+1}(h_k). \tag{2.53}$$

这公式虽然是在 $m\geqslant N$ 的假定下推出来的，但对 $m<N$ 的情形也仍然有用. 这是因为，当 $m<N$ 时令 $n\leqslant m$，(2.27)可改写为

$$T(h)=\tau_0+\tau_1\psi_1(h)+\cdots+\tau_n\psi_n(h)+\tau_{n+1}(h)\psi_{n+1}(h),$$

其中

$$\begin{aligned}\tau_{n+1}(h)&=\{\tau_{n+1}\psi_{n+1}(h)+\tau_{n+2}\psi_{n+2}(h)+\cdots\}/\psi_{n+1}(h)\\&=\tau_{n+1}+O(\psi_{n+2}(h)/\psi_{n+1}(h)).\end{aligned} \tag{2.54}$$

于是，把 n 看成 N，则由(2.53)得

$$T_m^{(i)}-\tau_0=\sum_{k=i}^{i+m}C_{mk}^{(i)}\{\tau_{n+1}+O(\psi_{n+2}(h_k)/\psi_{n+1}(h_k)\}\psi_{n+1}(h_k). \tag{2.55}$$

特别地，取 $n=m$，便有

$$T_m^{(i)}-\tau_0=\sum_{k=i}^{i+m}C_{mk}^{(i)}\{\tau_{m+1}+O(\psi_{m+2}(h_k)/\psi_{m+1}(h_k)\}\psi_{m+1}(h_k). \tag{2.56}$$

这结果的形式跟(2.47)、(2.48)相似.

上面导出的误差估计式，实际上很难应用．这是因为，对于一个实际的问题，尽管可以肯定渐近展开式(2.27)存在，但 τ_{m+1} 的具体数值往往难以确定．这时较好的办法，是通过计算过程中得到的量来估计误差．显然，如果能找到夹住极限 $T(0)$ 的两个趋于 $T(0)$ 的量，则将它们的中间值取为 $T(0)$ 的近似值时，其误差不会超过它们之差绝对值的一半．这两个量称为 $T(0)$ 的**渐近上、下限**．

对于多项式外推算法，由误差估计式(2.47)可见，若 m 固定($m<N$)且 i 充分大，则当 i 增加时，$T_m^{(i)}$ 单调地趋于 $T(0)$，可作为 $T(0)$ 的渐近上限或下限．为了求出由 $T(0)$ 另一侧单调地趋于 $T(0)$ 的 $U_m^{(i)}$，可令

$$U_m^{(i)} \equiv (1+\alpha)T_m^{(i+1)} - \alpha T_m^{(i)}, \qquad (2.57)$$

使 $U_m^{(i)} - T(0)$ 与(2.47)的符号相反，即

$$U_m^{(i)} - T(0) = (-1)^{m+1} h_i^r h_{i+1}^r \cdots h_{i+m}^r \{\tau_{m+1} + o(h_i^r)\}.$$

由于(2.57)和(2.47)，

$$U_m^{(i)} - T(0) = (-1)^m h_i^r h_{i+1}^r \cdots h_{i+m}^r \times \left\{\tau_{m+1}\left((1+\alpha)\frac{h_{i+m+1}^r}{h_i^r} - \alpha\right) + o(h_i^r)\right\},$$

这只需 α 满足

$$(1+\alpha)h_{i+m+1}^r / h_i^r - \alpha = -1,$$

即

$$\alpha = 1 + \frac{2}{(h_i/h_{i+m+1})^r - 1}. \qquad (2.58)$$

当 α 满足此条件时，$T_m^{(i)}$ 和 $U_m^{(i)}$ 是 $T(0)$ 的渐近上下限．故

$$\left|\frac{1}{2}(T_m^{(i)} + U_m^{(i)}) - T(0)\right| \leqslant \frac{1}{2}\left|T_m^{(i)} - U_m^{(i)}\right|. \qquad (2.59)$$

由(2.57)、(2.58)和(2.14)，可知

$$\frac{1}{2}(T_m^{(i)} + U_m^{(i)}) = T_m^{(i+1)} + \frac{T_m^{(i+1)} - T_m^{(i)}}{(h_i/h_{i+m+1})^r - 1} = T_{m+1}^{(i)},$$

$$\frac{1}{2}(T_m^{(i)}-U_m^{(i)})=\frac{h_i^r}{h_{i+m+1}^r-h_i^r}(T_m^{(i+1)}-T_m^{(i)})$$
$$=\frac{T_m^{(i+1)}-T_m^{(i)}}{(h_{i+m+1}/h_i)^r-1},$$

故(2.59)可改写为

$$|T_{m+1}^{(i)}-T(0)|\leqslant\frac{|T_m^{(i+1)}-T_m^{(i)}|}{1-(h_{i+m+1}/h_i)^r}=|T_{m+1}^{(i)}-T_m^{(i)}|. \tag{2.60}$$

如果在(2.57)中直接令 $\alpha=1$, 即令

$$U_m^{(i)}=2T_m^{(i+1)}-T_m^{(i)}, \tag{2.61}$$

则由(2.47)有

$$U_m^{(i)}-T(0)=(-1)^m h_i^r\cdots h_{i+m}^r$$
$$\times\left\{\tau_{m+1}\left(\frac{2h_{i+m+1}^r}{h^r}-1\right)+o(h_i^r)\right\}.$$

由此可见,如果

$$2(h_{i+m+1}/h_i)^r-1<0, \tag{2.62}$$

则 $T_m^{(i)}$、$U_m^{(i)}$ 仍然是 $T(0)$ 的渐近上下限,(2.59)仍然成立. 但这时

$$\frac{1}{2}(T_m^{(i)}+U_m^{(i)})=T_m^{(i+1)},$$
$$\frac{1}{2}(U_m^{(i)}-T_m^{(i)})=T_m^{(i+1)}-T_m^{(i)},$$

故(2.59)变为

$$|T_m^{(i+1)}-T(0)|\leqslant|T_m^{(i+1)}-T_m^{(i)}|. \tag{2.63}$$

用此公式估计误差实际上很保守. 因为,由(2.47)

$$T_m^{(i+1)}-T(0)=(-1)^m h_{i+1}^r\cdots h_{i+m+1}^r\{\tau_{m+1}+o(h_{i+1}^r)\},$$
$$T_m^{(i+1)}-T_m^{(i)}=(-1)^m h_{i+1}^r\cdots h_{i+m+1}^r$$
$$\times\left\{\tau_{m+1}\left(1-\frac{h_i^r}{h_{i+m+1}^r}\right)+o(h_i^r)\right\},$$

所以

$$|T_m^{(i+1)}-T(0)|\approx\frac{|T_m^{(i+1)}-T_m^{(i)}|}{(h_i/h_{i+m+1})^r-1}=|T_{m+1}^{(i)}-T_m^{(i+1)}|. \tag{2.64}$$

公式(2.60)、(2.63)、(2.64)表明，当 i 充分大时，外推表中某元素作为 $T(0)$ 近似值的误差，可用该元素与同一对角线上(或同一列上或同一行上)前一元素之差来估计. 第1章§4在讲到应用外推法时，通常发现同行或同列元素相差很小时便停止计算，其理论根据就在这里.

什么样的 i 才算"充分大"呢？这就是(2.47)中 $o(h_i^r)$ 可以忽略. 此时 $T_m^{(i)}$ 随 i 的增加而单调减或单调增，而

$$\begin{aligned}
D_m^{(i)}&\equiv\left(\frac{h_i}{h_{i+m+1}}\right)^r\frac{T_{m+1}^{(i)}-T_{m+1}^{(i+1)}}{T_{m+1}^{(i-1)}-T_m^{(i)}}\\
&=\left(\frac{h_i}{h_{i+m}}\right)^r\frac{h_{i+m}^r-h_{i-1}^r}{h_{i+m+1}^r-h_i^r}\cdot\frac{T_m^{(i+1)}-T_m^{(i)}}{T_m^{(i)}-T_m^{(i-1)}}\\
&=\left(\frac{h_i}{h_{i+m}}\right)^r\frac{h_{i+m}^r-h_{i-1}^r}{h_{i+m+1}^r-h_i^r}\\
&\quad\times\frac{(-1)^m h_{i+1}^r\cdots h_{i+m}^r\{\tau_{m+1}(h_{i+m+1}^r-h_i^r)+o(h_i^r)\}}{(-1)^m h_i\cdots h_{i+m-1}^r\{\tau_{m+1}(h_{i+m}^r-h_{i-1}^r)+o(h_{i-1}^r)\}}\\
&\approx 1.
\end{aligned} \tag{2.65}$$

实用中，$T_m^{(i)}$ 单调或 $D_m^{(i)}\approx1$，就算是 i 充分大的标志.

对于 Bulirsch-Stoer 外推算法，根据(2.48)也可作类似的讨论，只是公式(2.58)、(2.60)、(2.62)、(2.64)、(2.65)应分别修改为

$$\alpha=1+\frac{2}{b^{-r_{m+1}}-1}, \tag{2.58'}$$

$$|T_{m+1}^{(i)}-T(0)|\leqslant\frac{1}{1-b^{r_{m+1}}}|T_m^{(i+1)}-T_m^{(i)}|, \tag{2.60'}$$

$$2b^{r_{m+1}}<1, \tag{2.62'}$$

$$|T_m^{(i+1)}-T(0)|\approx\frac{|T_m^{(i+1)}-T_m^{(i)}|}{b^{-r_{m+1}}-1}=|T_{m+1}^{(i)}-T_m^{(i+1)}|,$$

(2.64′)

$$D_m^{(i)}\equiv\frac{T_m^{(i+1)}-T_m^{(i)}}{T_m^{(i)}-T_m^{(i-1)}}b^{-r_{m+1}}\approx 1. \qquad (2.65')$$

在实际问题中，有时可以肯定 $T(h)$ 具有(2.16)形式的渐近展开式，但其中的指数 r_1、r_2、$\cdots$ 却难以确定．这时由(2.65′)可见，当 i 充分大时

$$b^{r_{m+1}}\approx\frac{T_m^{(i+1)}-T_m^{(i)}}{T_m^{(i)}-T_m^{(i-1)}}.$$

根据此式我们可以确定 r_1、r_2、$\cdots$．当然，由右边分式是否随 i 增大而趋于定值，也可判断(2.16)形式的渐近展开式是否确实成立．

§6 收敛性与稳定性

对于多项式外推算法和 Bulirsch-Stoer 外推算法，由公式(2.47)、(2.48)可见，当 m 固定且小于渐近展开式(2.12)、(2.16)中的 N 时，

$$\lim_{i\to\infty}T_m^{(i)}=T(0). \qquad (2.66)$$

这表明，外推表中的 $m(m<N)$ 列元素收敛于 $T(0)$．

其实，在更一般的条件下，即对多项式外推法假定 $h_{i+1}/h_i\leqslant b<1$，对 Bulirsch-Stoer 算法假定级数

$$\sum_{j=1}^{\infty}b^{r_j}$$

收敛，我们不但可以证明列收敛，即(2.66)成立，而且还可以证明对角线收敛，即对固定的 i，

$$\lim_{m\to\infty}T_m^{(i)}=T(0). \qquad (2.67)$$

当然，由于外推算法不过用来加速 $\{T(h_i)\}$ 的收敛，所以序列 $\{T(h_i)\}$ 收敛于 $T(0)=\tau_0$ 应是自然的假定.

先证列收敛. 由(2.49)(a)和(b)可知

$$T_m^{(i)}=A_m^i T(h)=\sum_{k=i}^{i+m} C_{mk}^{(i)} T(h_k), \tag{2.68}$$

$$\sum_{k=i}^{i+m} C_{mk}^{(i)}=A_m^i 1=1. \tag{2.69}$$

由此，如果能证

$$\sum_{k=i}^{i+m} |C_{mk}^{(i)}| \leqslant M, \quad M \text{ 为常数}, \tag{2.70}$$

便可证明(2.66). 这是因为，对于任意 $\varepsilon>0$，由 $\{T(h_i)\}$ 的收敛性知，一定存在 N，使当 $i>N$ 时 $|T(h_i)-T(0)|<\varepsilon/M$，从而

$$\begin{aligned}|T_m^{(i)}-T(0)| &= \left|\sum_{k=i}^{i+m} C_{mk}^{(i)} T(h_k)-\sum_{k=i}^{i+m} C_{mk}^{(i)} T(0)\right| \\ &\leqslant \sum_{k=i}^{i+m} |C_{mk}^{(i)}|\,|T(h_k)-T(0)|<\sum_{k=i}^{i+m} |C_{mk}^{(i)}|\frac{\varepsilon}{M}\leqslant\varepsilon.\end{aligned}$$

所以关键在于证明(2.70).

对于多项式外推法，由(2.52)知

$$\begin{aligned}\sum_{k=i}^{i+m} |C_{mk}^{(i)}| &= \sum_{k=i}^{i+m} \prod_{j=i}^{i+m}{}' \frac{h_j^r}{|h_j^r-h_k^r|}=\sum_{k=0}^{m} \prod_{j=i}^{i+m}{}' \frac{h_j^r}{|h_j^r-h_{i+m-k}^r|} \\ &= \sum_{k=0}^{m} \prod_{j=i}^{i+m-k-1} \frac{1}{1-h_{i+m-k}^r/h_j^r} \\ &\quad \times \prod_{j=i+m-k+1}^{i+m} \frac{1}{h_{i+m-k}^r/h_j^r-1} \\ &\leqslant \sum_{k=0}^{m} \prod_{j=1}^{m-k} \frac{1}{1-b^{jr}} \times \prod_{j=1}^{k} \frac{1}{b^{-jr}-1} \\ &< \prod_{j=1}^{\infty} \frac{1}{1-b^{jr}} \sum_{k=0}^{\infty} \prod_{j=1}^{k} \frac{1}{b^{-jr}-1}=M_1.\end{aligned} \tag{2.71}$$

说明(2.70)确实成立.

对于 Bulirsch-Stoer 算法，我们在推导(2.18)时曾经看到，$C_{m,k+i}^{(i)}$ 是 $Q_m^{(i)}(x)$ 中 x^k 的系数，即

$$Q_m^{(i)}(x)=\sum_{k=0}^{m} C_{m,i+k}^{(i)} x^k.$$

而由(2.21)，

$$Q_m^{(i)}(x)=\prod_{j=1}^{m}(x-b^{r_j})\Big/\prod_{j=1}^{m}(1-b^{r_j}).$$

因此，

$$\sum_{k=i}^{i+m}|C_{mk}^{(i)}|=\sum_{k=0}^{m}|C_{m,i+k}^{(i)}|\leqslant\prod_{j=1}^{m}(1+b^{r_j})\Big/\prod_{j=1}^{m}(1-b^{r_j})$$

$$<\prod_{j=1}^{\infty}(1+b^{r_j})\Big/\prod_{j=1}^{\infty}(1-b^{r_j})=M_2. \qquad (2.72)$$

故对 Bulirsch-Stoer 算法，(2.70)也成立。

这样，我们便完成了(2.66)的证明。

为了证明对角线的收敛性，我们需要引用 **Toeplitz 定理**，即：若数列 $\{x_i\}$ 收敛于 τ_0，数组 $\{C_{mk}\}$ 满足以下三条件：

(a) 对所有 m，$\sum_{k=0}^{m} C_{mk}=1$；

(b) 对所有 m，存在常数 M，使 $\sum_{k=0}^{m}|C_{mk}|<M$；

(c) 对固定的 k，$\lim_{m\to\infty} C_{mk}=0$，

那么，若令 $\tilde{x}_m=\sum_{k=0}^{m} C_{mk}x_k$，则数列 $\{\tilde{x}_m\}$ 也收敛于 τ_0。

此定理证明如下：任给 $\varepsilon>0$，由于 $\{x_i\}$ 收敛于 τ_0，一定存在 N，使当 $m>N$ 时 $|x_m-\tau_0|<\varepsilon/2M$。由此，根据条件(a)，

$$|\tilde{x}_m-\tau_0|=\left|\sum_{k=0}^{m} C_{mk}x_k-\sum_{k=0}^{m} C_{mk}\tau_0\right|$$

$$\leqslant\sum_{k=0}^{N}|C_{mk}(x_k-\tau_0)|+\sum_{k=N+1}^{m}|C_{mk}|\,|x_k-\tau_0|$$

$$<\sum_{k=0}^{N}|C_{mk}(x_k-\tau_0)|+\frac{\varepsilon}{2M}\sum_{k=N+1}^{m}|C_{mk}|.$$

根据条件(b),

$$\frac{\varepsilon}{2M}\sum_{k=N+1}^{m}|C_{mk}|<\frac{\varepsilon}{2M}\cdot M=\frac{\varepsilon}{2}.$$

而根据条件(c),因 N 固定,可选取 $N_1>N$, 使 $m>N_1$ 时

$$\sum_{k=0}^{N}|C_{mk}(x_k-\tau_0)|<\frac{\varepsilon}{2}.$$

于是, $m>N_1$ 时

$$|\tilde{x}_m-\tau_0|<\frac{\varepsilon}{2}+\frac{\varepsilon}{2}=\varepsilon.$$

Toeplitz 定理证毕.

利用 Toeplitz 定理, 很容易证明对角线的收敛性, 即(2.67). 事实上, 把 $\{T(h_i)\}$ 看作 $\{x_i\}$, $\{C_{m,i+k}^{(i)}\}$ 看作 $\{C_{mk}\}$, 则由(2.69)和(2.70)知

$$\sum_{k=0}^{m}C_{m,i+k}^{(i)}=\sum_{k=i}^{i+m}C_{mk}^{(i)}=1,$$

$$\sum_{k=0}^{m}|C_{m,i+k}^{(i)}|=\sum_{k=i}^{i+m}|C_{mk}^{(i)}|<M,$$

说明条件(a),(b)成立. 当 k 固定时, 对于多项式外推算法, 由(2.52)知, 当 $m\to\infty$ 时

$$\frac{C_{m+1,k+i}^{(i)}}{C_{m,k+i}^{(i)}}=\frac{h_{i+m+1}^{r}}{h_{i+k}^{r}-h_{i+m+1}^{r}}\to 0,$$

可见 $\lim\limits_{m\to\infty}C_{m,i+k}^{(i)}=0$. 对于 Bulirsch-Stoer 外推算法, 我们在推导(2.18)时曾经指出, $C_{m,i+k}^{(i)}$ 是多项式

$$Q_m^{(i)}(x)=\sum_{k=0}^{m}C_{m,i+k}^{(i)}x^k=\prod_{j=1}^{m}\frac{x-b^{r_j}}{1-b^{r_j}}$$

中 x^k 的系数. 由多项式根与系数的关系(韦达定理)知, $\left|C_{m,i+k}^{(i)}\prod\limits_{j=1}^{m}(1-b^{r_j})\right|$ 应是 $Q_m^{(i)}(x)$ 的 m 个根 b^{r_1}, b^{r_2}, …, b^{r_m} 中取 $m-k$ 个根的乘积之和. 这种乘积不超过 $b^{(m-k)r_1}$, 而乘积

个数是牛顿二项式系数

$$\binom{m}{m-k}=\frac{m!}{(m-k)!k!}=\frac{m(m-1)\cdots(m-k+1)}{k!}\leqslant m^k.$$

故
$$\left|C_{m,i+k}^{(i)}\right|<m^k b^{(m-k)r_1}\Big/\prod_{j=1}^{m}(1-b^{r_j})$$
$$<m^k b^{mr_1}\Big/b^{kr_1}\prod_{j=1}^{\infty}(1-b^{r_j}).$$

当 $m\to\infty$ 时，因 $m^k b^{mr_1}\to 0$，便知 $\left|C_{m,i+k}^{(i)}\right|\to 0$. 这样，对于两种外推算法，条件(c)都成立. 于是根据 Toeplitz 定理，(2.67)成立. 证毕.

应当注意，我们证明外推表中列和对角线的收敛性时，没有涉及到 $T(h)$ 的渐近展开式. 因此，不管渐近展开式(2.12)、(2.16)是否成立，只要 $\{T(h_i)\}$ 收敛于 τ_0，按照多项式外推算法(假定 $h_{i+1}/h_i\leqslant b<1$)或 Bulirsch-Stoer 外推算法(假定 $\sum\limits_{j=1}^{\infty}b^{r_j}$ 收敛)构造外推表 1，每一列或每一对角线都收敛于 τ_0. 不过，这是否加速了 $\{T(h_i)\}$ 的收敛，那就难说了.

在渐近展开式(2.12)、(2.16)成立的情况下，多项式外推法与 Bulirsch-Stoer 外推法会大大加速 $\{T(h_i)\}$ 的收敛性，则是毫无疑问的. 这是因为，对固定的 i，当 m 增加时，如果 m 不超过渐近展开式中的 N，误差的阶数会逐步提高. 而对固定的 m，由误差估计式(2.47)和(2.48)可见

$$T_m^{(i)}-T(0)=O(h_i^r\cdots h_{i+m}^r)\quad \text{或}\quad O(h_i^{r_{m+1}}b^{\sum_{j=1}^{m}r_j}).$$

应当注意，对固定的 i，如果 $m\geqslant N$，则由误差估计式(2.53)可见，当 m 增加时，$T_m^{(i)}$ 的误差阶数不会再提高. 正是由于这个原因，当渐近展开式中的 N 为有限值时，计算 $T_m^{(i)}(m>N)$ 的意义不大. 如果渐近展开式(2.12)、(2.16)根本不成立，可设想此时(2.12)、(2.16)中的 $N=0$，$\tau_{N+1}(h)$ 数

值很大,因而(2.53)右方很大,多项式外推法与 Bulirsch-Stoer 外推法就不能加速 $\{T(h_i)\}$ 的收敛了.

从公式(2.70)还可得出有关多项式外推算法与 Bulirsch-Stoer 外推算法稳定性的推论. 事实上, 设 $\tilde{T}(h_i)$ 是 $T(h_i)$ 的近似值, $\tilde{T}_m^{(i)}$ 是由 $\tilde{T}(h_i)$ 出发按这两种外推算法算出的结果, 则根据(2.68)和(2.70)有

$$\begin{aligned}\left|\tilde{T}_m^{(i)}-T_m^{(i)}\right| &= \left|\sum_{k=i}^{i+m} C_{mk}^{(i)}\tilde{T}(h_k)-\sum_{k=i}^{i+m} C_{mk}^{(i)}T(h_k)\right| \\ &\leqslant \sum_{k=i}^{i+m}\left|C_{mk}^{(i)}\right|\left|\tilde{T}(h_k)-T(h_k)\right| \\ &\leqslant \max_k\left|\tilde{T}(h_k)-T(h_k)\right|\sum_{k=i}^{i+m}\left|C_{mk}^{(i)}\right| \\ &\leqslant M\cdot\max_k\left|\tilde{T}(h_k)-T(h_k)\right|. \qquad (2.73)\end{aligned}$$

这表明, M 越大,这两种外推算法越不稳定; M 越小,则越稳定. 由(2.71)和(2.72)可见, b 越小 M 越小; b 越接近 1, M 就越大,算法就越不稳定.

现将 Romberg 算法的 M 之值列表如下:

表 4

M \ h_i / m	$h_0/(1+i)$	$h_0\begin{cases}2^{-\frac{i-1}{2}}\\3^{-1}2^{-\frac{i-2}{2}}\end{cases}$	$2^{-i}h_0$	$10^{-i}h_0$	$(0.9)^i h_0$
1	1.67	1.67	1.67	1.02	9.53
2	3.13	3.13	1.89	1.02	45.88
5	26.44	6.43	1.97	1.02	780
6	55.82	7.37	1.97	1.02	1394
12	5730	7.74	1.97	1.02	7980
14	27670	7.75	1.97	1.02	10086

第 **3** 章

有理式外推法

§1 有理式插值的概念

广义多项式外推法，是用广义插值多项式 $P_m^{(i)}(h)$ 近似替代 $T(h)$，用 $P_m^{(i)}(0)$ 推算 $T(0)$. 由此自然想到，如果 $T(h)$ 用有理函数(有理分式)逼近较好，那就应当用有理插值函数 $R_{\mu\nu}^{(i)}(h)$ 来近似替代 $T(h)$，用 $R_{\mu\nu}^{(i)}(0)$ 来推算 $T(0)$. 这种推算 $T(0)=\tau_0$ 的方法称为**有理式外推法**。

所谓有理插值函数 $T_m^{(i)}(h)\equiv R_{\mu\nu}^{(i)}(h)\ (m=\mu+\nu)$，就是满足插值条件

$$T_m^{(i)}(h_k)\equiv R_{\mu\nu}^{(i)}(h_k)=T(h_k)\quad (k=i,\ i+1,\ \cdots,\ i+m). \tag{3.1}$$

的分式，其分子、分母分别是 h 次数不超过 μ、ν 的多项式，即

$$R_{\mu\nu}^{(i)}(h)=P_\mu^{(i)}(h)/Q_\nu^{(i)}(h),$$

$$P_\mu^{(i)}(h)=a_0^{(i)}+a_1^{(i)}h+\cdots+a_\mu^{(i)}h^\mu,$$

$$Q_\nu^{(i)}(h)=b_0^{(i)}+b_1^{(i)}h+\cdots+b_\nu^{(i)}h^\nu,$$

其中 $b_0^{(i)}$、$b_1^{(i)}$、$\cdots$、$b_\nu^{(i)}$ 不全为零. $R_{\mu\nu}^{(i)}(h)$ 常称为 μ/ν 次有理分式. 由于分式的分子、分母同乘一非零的常数而值不变，$R_{\mu\nu}^{(i)}(h)$ 中任一非零的系数可取成固定的数，例如1(否则分子、分母同除以这系数即可)，所以 $R_{\mu\nu}^{(i)}(h)$ 的 $\mu+\nu+2$ 个系数实质上只有 $\mu+\nu+1=m+1$ 个是独立的. 根据插值条件(3.1)，确定这些系数的方程组为

$$a_0^{(i)}+a_1^{(i)}h_k+\cdots+a_\mu^{(i)}h_k^\mu=T(h_k)(b_0^{(i)}+b_1^{(i)}h_k+\cdots+b_\nu^{(i)}h_k^\nu)$$
$$(k=i,\ i+1,\ \cdots,\ i+m). \tag{3.2}$$

这方程组不一定有解，所以满足插值条件(3.1)的插值有理函数 $R_{\mu\nu}^{(i)}(h)$ 不一定存在．但是，如果存在，则有

$$a_0^{(i)}+a_1^{(i)}h+\cdots+a_\mu^{(i)}h^\mu$$
$$=R_{\mu\nu}^{(i)}(h)(b_0^{(i)}+b_1^{(i)}h+\cdots+b_\nu^{(i)}h^\nu). \tag{3.3}$$

(3.2)和(3.3)一起，是以 $\mu+\nu+2$ 个系数为未知数的线性齐次方程组，它有非零解的充要条件是

$$\begin{vmatrix} 1 & h_i & \cdots & h_i^\mu & T(h_i) & T(h_i)h_i & \cdots & T(h_i)h_i^\nu \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & h_{i+m} & \cdots & h_{i+m}^\mu & T(h_{i+m}) & T(h_{i+m})h_{i+m} & \cdots & T(h_{i+m})h_{i+m}^\nu \\ 1 & h & \cdots & h^\mu & R_{\mu\nu}^{(i)}(h) & R_{\mu\nu}^{(i)}(h)h & \cdots & R_{\mu\nu}^{(i)}(h)h^\nu \end{vmatrix}=0. \tag{3.4}$$

将行列式按最后一行展开，把包含 $R_{\mu\nu}^{(i)}(h)$ 的那些项的总和记为 $Q(h)R_{\mu\nu}^{(i)}(h)$，把不含 $R_{\mu\nu}^{(i)}(h)$ 的那些项的总和记为 $-P(h)$. 显然 $P(h)$、$Q(h)$ 是 h 的多项式，它们的次数分别不超过 μ 和 ν．于是得到

$$-P(h)+R_{\mu\nu}^{(i)}(h)Q(h)=0,\quad R_{\mu\nu}^{(i)}(h)=P(h)/Q(h).$$

这里 $P(h)$、$Q(h)$ 是完全确定的，说明如果满足插值条件(3.1)的有理函数存在的话（即 $Q(h)\neq 0$），便是完全确定的.

通过计算行列式(3.4)计算 $R_{\mu\nu}^{(i)}(h)$ 是不方便的．特别是在改变有理分式的分子、分母次数，以提高近似精度的时候，尤其不方便．最好的办法，还是采用递推算法.

§2 连分式算法

为了导出计算插值有理分式的递推算法，常常采用连分

式. 所谓连分式,就是如下形式的分式:

$$\beta_0+\cfrac{\alpha_1}{\beta_1+\cfrac{\alpha_2}{\beta_2+\cfrac{\alpha_3}{\beta_3+\ddots+\cfrac{\alpha_n}{\beta_n}}}} \tag{3.5}$$

或为书写简便起见,记作

$$\beta_0+\frac{\alpha_1}{\beta_1}+\frac{\alpha_2}{\beta_2}+\frac{\alpha_3}{\beta_3}+\cdots+\frac{\alpha_n}{\beta_n}.$$

或

$$\beta_0+\frac{\alpha_1|}{|\beta_1}+\frac{\alpha_2|}{|\beta_2}+\frac{\alpha_3|}{|\beta_3}+\cdots+\frac{\alpha_n|}{|\beta_n}.$$

分式 α_k/β_k 称为连分式(3.5)的第 k 节分式。由 $k+1$ 节分式构成的连分式

$$\beta_0+\frac{\alpha_1}{\beta_1}+\frac{\alpha_2}{\beta_2}+\cdots+\frac{\alpha_k}{\beta_k}$$

称为连分式(3.5)的第 k 个渐近分式。它可化为普通分式 p_k/q_k. 事实上,

$$\frac{p_0}{q_0}=\frac{\beta_0}{1},\quad \frac{p_1}{q_1}=\frac{\beta_0\beta_1+\alpha_1}{\beta_1},\quad \frac{p_2}{q_2}=\frac{\beta_2p_1+\alpha_2p_0}{\beta_2q_1+\alpha_2q_0},$$

一般,如果

$$p_k=\beta_kp_{k-1}+\alpha_kp_{k-2},\quad q_k=\beta_kq_{k-1}+\alpha_kq_{k-2}, \tag{3.6}$$

则有

$$\frac{p_{k+1}}{q_{k+1}}=\beta_0+\frac{\alpha_1}{\beta_1}+\cdots+\cfrac{\alpha_k}{\beta_k+\cfrac{\alpha_{k+1}}{\beta_{k+1}}}$$

$$=\frac{\left(\beta_k+\frac{\alpha_{k+1}}{\beta_{k+1}}\right)p_{k-1}+\alpha_kp_{k-2}}{\left(\beta_k+\frac{\alpha_{k+1}}{\beta_{k+1}}\right)q_{k-1}+\alpha_kq_{k-2}}$$

$$=\frac{\beta_{k+1}(\beta_kp_{k-1}+\alpha_kp_{k-2})+\alpha_{k+1}p_{k-1}}{\beta_{k+1}(\beta_kq_{k-1}+\alpha_kq_{k-2})+\alpha_{k+1}q_{k-2}}=\frac{\beta_{k+1}p_k+\alpha_{k+1}p_{k-1}}{\beta_{k+1}q_k+\alpha_{k+1}q_{k-1}}.$$

这样，我们实际上用数学归纳法证明了递推公式(3.6)对一切 $k\geqslant 2$ 都成立. 它可用来计算连分式(3.5)各渐近分式 p_k/q_k 的值. 不过，在电子计算机上除法与乘法运算速度差不多，也可直接用除法计算渐近分式 p_k/q_k. 具体算法如下：

$$\begin{cases} r_k=\alpha_k/\beta_k, \\ r_j=\alpha_j/(\beta_j+r_{j+1}) \quad (j=k-1,\ k-2,\ \cdots,\ 1), \\ p_k/q_k=\beta_0+r_1. \end{cases}$$

我们用(3.6)，主要是为了理论分析.

从公式(3.6)可见，如果 $\beta_k(k>1)$ 是常数，α_k 是 h 的一次式，用 $\deg p_k$ 表示 $p_k(h)$ 的次数，那么

$$\deg p_k\leqslant \max(\deg p_{k-1},\ \deg p_{k-2}+1).$$

由此可见，当 p_{k-1} 的次数只比 p_{k-2} 高一次时，p_k 的次数不增加；只有 p_{k-1} 与 p_{k-2} 的次数相同时，p_k 的次数才会增加一次. 对于 q_k 也有类似的结论. 这样，如果插值有理分式 $T_m^{(i)}(h)=R_{\mu\nu}^{(i)}(h)$ 可表示为连分式

$$T_m^{(i)}(h)=P_l^{(i)}(h)+\frac{(h-h_i)(h-h_{i+1})\cdots(h-h_{i+l})}{C_1}$$
$$+\frac{h-h_{i+l+1}}{C_2}+\cdots+\frac{h-h_{i+m-1}}{C_{m-l}}. \tag{3.7}$$

其中 $P_l^{(i)}(h)$ 是次数为 l 的多项式. 当 $m=l,\ l+1,\ l+2$ 时 $T_m^{(i)}(h)$ 的次数分别为

$$l/0,\ l+1/0,\ l+1/1,\ l+2/1,\ l+2/2,$$
$$l+3/2,\ l+3/3,\ \cdots.$$

由于 $T_m^{(i)}(h)$ 满足插值条件(3.1)，由(3.7)显然

$$P_l^{(i)}(h_k)=T(h_k) \quad (k=i,\ i+1,\ \cdots,\ i+l).$$

这说明 $P_l^{(i)}(h)$ 是 $T(h)$ 的 l 次插值多项式，可按逐次线性插值公式(2.10)计算. 现在问题在于如何选择系数 $C_1,\ C_2,\ \cdots,$

C_{m-l}，使满足其余插值条件．将(3.7)改写为

$$\begin{cases} T_m^{(i)}(h)=P_l^{(i)}(h)+(h-h_i)\cdots(h-h_{i+l})/V_1(h), \\ V_1(h)=C_1+(h-h_{i+l+1})/V_2(h), \\ V_2(h)=C_2+(h-h_{i+l+2})/V_3(h), \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ V_{m-l-1}(h)=C_{m-l-1}+(h-h_{i+m-1})/V_{m-l}(h), \\ V_{m-l}(h)=C_{m-l}. \end{cases} \tag{3.8}$$

令 $h=h_{i+l+k}$, $k=1, 2, \cdots, m-l$, 则

$$\begin{cases} T(h_{i+l+k})=P_l^{(i)}(h_{i+l+k})+(h_{i+l+k}-h_i)\cdots \\ \quad (h_{i+l+k}-h_{i+l})/V_1(h_{i+l+k}), \\ V_k(h_{i+l+k})=C_k, \\ V_j(h_{i+l+k})=C_j+(h_{i+l+k}-h_{i+l+j})/V_{j+1}(h_{i+l+k}), \ j\neq k. \end{cases}$$

故当 $k=1, 2, \cdots, m-l$ 时

$$\begin{cases} V_1(h_{i+l+k})=\dfrac{(h_{i+l+k}-h_i)\cdots(h_{i+l+k}-h_{i+l})}{T(h_{i+l+k})-P_l^{(i)}(h_{i+l+k})}, \\ V_{j+1}(h_{i+l+k})=\dfrac{h_{i+l+k}-h_{i+l+j}}{V_j(h_{i+l+k})-C_j} \quad (j=1, 2, \cdots, k-1), \\ C_k=V_k(h_{i+l+k}). \end{cases} \tag{3.9}$$

这便是计算(3.7)中系数 $C_1, C_2, \cdots, C_{m-l}$ 的递推公式．

有了这些系数，按照(3.8)倒推，可以算出插值有理分式 $T_m^{(i)}(h)$的值，特别是 $T_m^{(i)}(0)$的值．把 $T_m^{(2)}(0)$仍记为 $T_m^{(i)}$．根据这些公式推算 $T(0)$近似值 $T_m^{(i)}$ 的方法，称为 **Stoer 连分式外推法**．

在 $l=0$ 的情形，因为

$$P_0^{(i)}(h)=T(h_i),$$

令

$$C_0=T(h_0),$$

$$V_0(h_{i+k})=T(h_{i+k}),$$

则当 $k=1, 2, \cdots, m$ 时(3.9)变为

$$\begin{cases} V_0(h_{i+k})=T(h_{i+k}), \\ V_{j+1}(h_{i+k})=\dfrac{h_{i+k}-h_{i+j}}{V_j(h_{i+k})-C_j} \quad (j=0,\ 1,\ \cdots,\ k-1), \\ C_k=V_k(h_{i+k}). \end{cases} \tag{3.10}$$

这里计算 C_k 的递推公式(3.10)，类似于(2.8)中计算差商 $C_k^{(i)}$ 的公式，只是计算 $V_{j+1}(h_{i+k})$的分式反转过来了. 所以 C_k 称为 **k 阶反差商**；而由此计算有理插值函数 $T_m^{(i)}(h)$之值的方法，称为**反差商有理插值法**，或 **Thiele 算法**.

在 $l=0$ 的情况下，当 m 为偶数时，(3.7)中插值有理函数 $T_m^{(i)}(h)=P_m(h)/Q_m(x)$的分子、分母次数相同，均为 $m/2$ 次. 由递推公式(3.6)还可推知，Q_m 的最高次项系数为 1，P_m 的最高次项系数为 $C_0+C_2+\cdots+C_m$. 于是

$$\lim_{h\to\infty} T_m^{(i)}(h)=\lim_{h\to\infty}\frac{P_m}{Q_m}=C_0+C_2+\cdots+C_m. \tag{3.11}$$

由此可见，计算 $T_m^{(i)}(\infty)$比计算 $T_m^{(i)}(0)$简单. 由于这个缘故，公式(3.10)和(3.11)常用来推算 $T(\infty)$. 这称为 **Wuytack 算法**. 它是 1971 年 L. Wuytack 提出来的.

§3 Larkin 逐步插值法

上面计算插值有理函数 $T_m^{(i)}(h)$的过程要先算辅助量 C_k. 这种算法类似于 Newton 差商插值公式. 类似于 Neville 逐步线性插值公式的，则是 1967 年 Larkin 提出的递推算法.

当 $\mu=0$ 或 $\nu=0$ 时，有理插值函数 $T_m^{(i)}(h)=R_{\mu\nu}^{(i)}(h)$是容易计算的. 这是因为，当 $\nu=0$ 时，$T_m^{(i)}(h)$ (下面用 $T_m^{(i)}$ 表示)是多项式，故按 Neville 插值公式(2.10)，

$$T_m^{(i)}=\frac{(h-h_i)T_{m-1}^{(i+1)}-(h-h_{i+m})T_{m-1}^{(i)}}{h_{i+m}-h_i}. \tag{3.12}$$

当 $\mu=0$ 时, $1/T_m^{(i)}$ 是多项式, 故由上式

$$T_m^{(i)}=\frac{h_{i+m}-h_i}{(h-h_i)T_{m-1}^{(i+1)}-(h-h_{i+m})}$$

$$=\frac{h_{i+m}-h_i}{\dfrac{h-h_i}{T_{m-1}^{(i+1)}}+\dfrac{h_{i+m}-h}{T_{m-1}^{(i)}}}. \tag{3.13}$$

为了计算 μ 和 ν 均不为零时的有理函数 $T_m^{(i)}(h)$, Larkin 提出的递推算法如下: 令 $T_0^{(i)}=T(h_i)$, 当 $1\leqslant m\leqslant l$ 时按 (3.12) 或 (3.13) 计算, 当 $m>l$ 时则按下式计算:

$$T_m^{(i)}=T_{m-2}^{(i+1)}+\frac{h_{i+m}-h_i}{\dfrac{h-h_i}{T_{m-1}^{(i+1)}-T_{m-2}^{(i+1)}}+\dfrac{h_{i+m}-h}{T_{m-1}^{(i)}-T_{m-2}^{(i+1)}}}. \tag{3.14}$$

根据使用(3.12)或(3.13)的不同, 分别称之为 A_l 算法或 B_l 算法. 显然, 这样得出的 $T_m^{(i)}$ 是一个分式. 可以证明, 除特殊情况外, $T_m^{(i)}=P_m^{(i)}/Q_m^{(i)}$ 满足插值条件(3.1), 而且对算法 A_l,

$$\deg P_m^{(i)}\leqslant\left[\frac{m+1}{2}\right],\ \deg Q_m^{(i)}\leqslant\left[\frac{m-l+1}{2}\right], \tag{3.15}$$

其中第一式的等号当 $m-l$ 为偶数时成立, 第二式的等号当 $m-l$ 为奇数时成立, 又 $[x]$ 表示 x 的整数部分; 对 B_l 算法, 则有

$$\deg P_m^{(i)}\leqslant\left[\frac{m-l+1}{2}\right],\quad \deg Q_m^{(i)}\leqslant\left[\frac{m+1}{2}\right], \tag{3.16}$$

其中第一式的等号当 $m-l$ 为奇数时成立, 第二式的等号当 $m-l$ 为偶数时成立. 这样, 当 $m=0, 1, 2, \cdots, l, l+1, \cdots$ 时, A_l 算法产生的 $T_m^{(i)}$ 次数为

$$0/0,\ 1/0,\ 2/0,\ \cdots,\ l/0,\ l/1,\ l+1/1,$$
$$l+1/2,\ l+2/2,\ \cdots, \tag{3.17}$$

而 B_l 算法产生的 $T_m^{(i)}$ 次数为

$$0/0,\ 0/1,\ 0/2,\ \cdots,\ 0/l,\ 1/l,\ 1/l+1,$$
$$2/l+1,\ 2/l+2,\ \cdots. \qquad (3.17')$$

下面只就 A_l 算法来加以证明. B_l 算法的证明类似. 用数学归纳法. $m \leqslant l$ 时插值条件 (3.1) 和次数关系式 (3.15) 显然都成立. 现在假定 $T_{m-1}^{(i)}$、$T_{m-1}^{(i+1)}$、$T_{m-2}^{(i)}$ 满足条件(3.1)和(3.15). 设 h_r 是 $h_i, h_{i+1}, \cdots, h_{i+m}$ 中使 $T_m^{(i)}$ 不满足插值条件(3.1)的点, 即

$$T_m^{(i)}(h_r) = P_m^{(i)}(h_r)/Q_m^{(i)}(h_r) \neq T(h_r).$$

对所有这种点 h_r, 令 $E = \prod(h-h_r)$, 并令

$$\tilde{P}_m^{(i)} = EP_m^{(i)}, \quad \tilde{Q}_m^{(i)} = EQ_m^{(i)}. \qquad (3.18)$$

则递推公式(3.14)可改写为

$$\frac{h_{i+m}-h_i}{\dfrac{\tilde{P}_m^{(i)}}{Q_m^{(i)}} - \dfrac{P_{m-2}^{(i+1)}}{Q_{m-2}^{(i+1)}}} = \frac{h-h_i}{\dfrac{P_{m-1}^{(i+1)}}{Q_{m-1}^{(i+1)}} - \dfrac{P_{m-2}^{(i+1)}}{Q_{m-2}^{(i+1)}}} + \frac{h_{i+m}-h}{\dfrac{P_{m-1}^{(i)}}{Q_{m-1}^{(i)}} - \dfrac{P_{m-2}^{(i+1)}}{Q_{m-2}^{(i+1)}}}.$$

$$(3.19)$$

显然其中各分式的分母当 $h = h_{i+1}, h_{i+2}, \cdots, h_{i+m-1}$ 时都变为 0. 故令

$$z = \prod_{j=i+1}^{i+m-1}(h-h_s),$$

则必有

$$\tilde{P}_m^{(i)} Q_{m-2}^{(i+1)} - P_{m-2}^{(i+1)} \tilde{Q}_m^{(i)} = UZ, \qquad (3.20)$$

$$P_{m-1}^{(i)} Q_{m-2}^{(i+1)} - P_{m-2}^{(i+1)} Q_{m-1}^{(i)} = VZ, \qquad (3.21)$$

$$P_{m-1}^{(i+1)} Q_{m-2}^{(i+1)} - P_{m-2}^{(i+1)} Q_{m-1}^{(i+1)} = WZ. \qquad (3.22)$$

这里 U、V、W 中若有一个恒等于 0, 则由(3.19)可推出 $T_m^{(i)}$、$T_{m-1}^{(i)}$、$T_{m-1}^{(i+1)}$、$T_{m-2}^{(i+1)}$ 中至少有三个彼此恒等. 对于这种特殊情况, 我们将在以后考虑. 现在假定 U、V、W 均不恒等于 0. 此时(3.21)和(3.22)右边 h 的次数至少为 $m-1$. 而左边的次数, 由(3.15)知, 不超过

$$\max\left\{\left[\frac{m+l}{2}\right]+\left[\frac{m-l-1}{2}\right],\left[\frac{m+l-2}{2}\right]+\left[\frac{m-l+1}{2}\right]\right\}$$
$$=m-1.$$

由此可见, V 和 W 应为常数. 将(3.19)中各分母分别通分, 并约去公因式 $Q_{m-2}^{(i+1)}/Z$, 得

$$(h_{i+m}-h_i)\frac{\widetilde{Q}_m^{(i)}}{U}=(h-h_i)\frac{Q_{m-1}^{(i+1)}}{W}+(h_{i+m}-h)\frac{Q_{m-1}^{(i)}}{V}. \tag{3.23}$$

将(3.20)、(3.21)、(3.22)分别乘以 $(h_{i+m}-h_i)/U$, $(h_{i+m}-h)/V$, $(h-h_i)/W$, 然后相加, 并注意(3.23), 得

$$(h_{i+m}-h_i)\frac{\widetilde{P}_m^{(i)}}{U}=(h-h_i)\frac{P_{m-1}^{(i+1)}}{W}+(h_{i+m}-h)\frac{P_{m-1}^{(i)}}{V}. \tag{3.24}$$

以(3.23)除(3.24), 得

$$T_m^{(i)}=\frac{\widetilde{P}_m^{(i)}}{\widetilde{Q}_m^{(i)}}=\frac{(h-h_i)\dfrac{P_{m-1}^{(i+1)}}{W}+(h_{i+m}-h)\dfrac{P_{m-1}^{(i)}}{V}}{(h-h_i)\dfrac{Q_{m-1}^{(i+1)}}{W}+(h_{i+m}-h)\dfrac{Q_{m-1}^{(i)}}{V}}. \tag{3.25}$$

在此式中令 $h=h_i, h_{i+1}, \cdots, h_{i+m}$, 右边得到 $T(h)$ 相应的值. 这说明 $T_m^{(i)}$ 满足插值条件(3.1), 且 $E=1$, $\widetilde{P}_m^{(i)}=P_m^{(i)}$, $\widetilde{Q}_m^{(i)}=Q_m^{(i)}$.

注意(3.23)和(3.24)的右边, 它们都是 h 的多项式, 因此 U 应整除 $P_m^{(i)}$ 和 $Q_m^{(i)}$. 但因 $P_m^{(i)}$ 和 $Q_m^{(i)}$ 没有公因式, 所以 U 也应为常数.

由(3.21)和(3.22)消去 z, 得

$$Q_{m-2}^{(i+1)}\left(\frac{P_{m-1}^{(i+1)}}{W}-\frac{P_{m-1}^{(i)}}{V}\right)=P_{m-2}^{(i+1)}\left(\frac{Q_{m-1}^{(i+1)}}{W}-\frac{Q_{m-1}^{(i)}}{V}\right). \tag{3.26}$$

而考察(3.21)、(3.22)两边 h 的次数知

$$m-1\leqslant\max\left\{\left[\frac{m+l-1}{2}\right]+\deg Q_{m-2}^{(i+1)},\right.$$

$$\left.\deg P_{m-2}^{(i+1)}+\left[\frac{m-l}{2}\right]\right\}. \tag{3.27}$$

当 $m-l$ 为奇数时, $m+l$ 也是奇数, 上式变为

$$m-1\leqslant\max\left\{\frac{l+m-1}{2}+\deg Q_{m-2}^{(i+1)},\right.$$

$$\left.\frac{l+m-3}{2}+\frac{m-l-1}{2}(=m-2)\right\}.$$

所以

$$\deg Q_{m-2}^{(i+1)}=\frac{m-l-1}{2}.$$

由此, 按(3.26)

$$\frac{1}{2}(m-l-1)+\deg\left(\frac{P_{m-1}^{(i+1)}}{W}-\frac{P_{m-1}^{(i)}}{V}\right)$$

$$\leqslant\frac{1}{2}(m+l-3)+\frac{1}{2}(m-l-1).$$

从而

$$\deg\left(\frac{P_{m-1}^{(i+1)}}{W}-\frac{P_{m-1}^{(i)}}{V}\right)\leqslant\frac{m+l-3}{2}=\left[\frac{m+l-2}{2}\right], \tag{3.28}$$

于是由(3.24)得

$$\deg P_{m}^{(i)}\leqslant\max\left\{\frac{m+l-3}{2}+1,\left[\frac{m+l-1}{2}\right]\right\}$$

$$=\left[\frac{m+l}{2}\right].$$

类似地, 当 $m-l$ 为偶数时, $m+l$ 也是偶数, 由(3.27)可得 $\deg P_{m-2}^{(i+1)}=m-1$, 由(3.26)可得

$$\deg\left(\frac{Q_{m-1}^{(i+1)}}{W}-\frac{Q_{m-1}^{(i)}}{V}\right)\leqslant\left[\frac{m-l+1}{2}\right],$$

从而由(3.23)得到

$$\deg Q_m^{(i)} \leqslant \left[\frac{m-l+1}{2}\right].$$

这样我们证明了，在 U、V、W 均不恒等于 0 的情况下，$T_m^{(i)}$ 满足插值条件(3.1)和次数关系(3.15)．只是等号什么时候成立尚未研究．

在 U、V、W 中有一个恒等于 0 的情况下，由(3.19)到(3.22)可知，必有 $T_m^{(i)} \equiv T_{m-2}^{(i+1)} \equiv T_{m-1}^{(i)}$(或 $T_{m-1}^{(i+1)}$)．此时公式(3.15)自然成立，但插值条件(3.1)不一定成立．由于 $T_{m-2}^{(i+1)} \equiv T_{m-1}^{(i)}$ 或 $T_{m-2}^{(i+1)} \equiv T_{m-1}^{(i+1)}$，说明 $T_{m-2}^{(i+1)}$ 的 $m-1$ 个独立系数要满足 m 个方程(插值条件)．这类问题不一定有解，$T_m^{(i)}$ 不一定满足插值条件(3.1)就不足为怪了．当 $T_{m-1}^{(i)} \equiv T_{m-1}^{(i+1)}$ 时，$T_m^{(i)} \equiv T_{m-1}^{(i)}$ 满足插值条件(3.1)．

最后尚须证明，在 $T_m^{(i)}$ 满足插值条件的情况下，除非 $T_m^{(i)} \equiv T_{m-1}^{(i)} \equiv T_{m-1}^{(i+1)}$，公式(3.15)中两式的等号分别当 $m-l$ 为偶数、奇数时成立．事实上，考虑

$$Y \equiv P_m^{(i)} Q_{m-1}^{(i)} - P_{m-1}^{(i)} Q_m^{(i)}.$$

它有零点 $h_i, h_{i+1}, \cdots, h_{i+m-1}$．可见除非 $Y \equiv 0$(此时由(3.19)可知 $T_m^{(i)} \equiv T_{m-1}^{(i)} \equiv T_{m-1}^{(i+1)}$)，必有 $\deg Y \geqslant m$，从而

$$\max\{\deg P_m^{(i)} + \deg Q_{m-1}^{(i)},\ \deg P_{m-1}^{(i)} + \deg Q_m^{(i)}\} \geqslant m.$$

由此，当 $m-l$ 为偶数时

$$\max\left\{\deg P_m^{(i)} + \deg Q_{m-1}^{(i)},\ \frac{m+l-2}{2} + \frac{m-l}{2}(=m-1)\right\}$$

$$\geqslant m,$$

从而

$$\deg P_m^{(i)} = \frac{m+l}{2} = \left[\frac{m+l}{2}\right],$$

$$\deg Q_{m-1}^{(i)} = \frac{m-l}{2} = \left[\frac{m-l+1}{2}\right];$$

而当 $m-l$ 为奇数时

$$\max\left\{\frac{m+l-1}{2}+\frac{m-l-1}{2}(=m-1),\right.$$

$$\left.\deg P_{m-1}^{(i)}+\deg Q_m^{(i)}\right\}\geqslant m,$$

从而
$$\deg Q_m^{(i)}=\frac{m-l+1}{2}=\left[\frac{m-l+1}{2}\right],$$

$$\deg P_{m-1}^{(i)}=\frac{m+l-1}{2}=\left[\frac{m+l}{2}\right].$$

证毕.

§4 有理式逐步外推法

在公式(3.12)～(3.14)中令 $h=0$, 立即得到计算 $T_m^{(i)}(0)$ 的如下递推公式, 其中 $T_m^{(i)}$ 表示 $T_m^{(i)}(0)$, 当然 $T_0^{(i)}=T(h_i)$.

$$T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{h_i/h_{i+m}-1},\tag{3.29}$$

$$T_m^{(i)}=\frac{h_i-h_{i+m}}{h_i/T_{m-1}^{(i+1)}-h_{i+m}/T_{m-1}^{(i)}},\tag{3.30}$$

$$T_m^{(i)}=T_{m-2}^{(i+1)}+\frac{h_i-h_{i+m}}{\dfrac{h_i}{T_{m-1}^{(i+1)}-T_{m-2}^{(i+1)}}-\dfrac{h_{i+m}}{T_{m-1}^{(i)}-T_{m-2}^{(i+1)}}}.\tag{3.31}$$

公式(3.30)和(3.31)也可分别改写为

$$T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\dfrac{h_i}{h_{i+m}}\left(1-\dfrac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{T_{m-1}^{(i+1)}}\right)-1},\tag{3.32}$$

$$T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\dfrac{h_i}{h_{i+m}}\left(1-\dfrac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{T_{m-1}^{(i+1)}-T_{m-2}^{(i+1)}}\right)-1}$$

$$=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\dfrac{h_i}{h_{i+m}}\theta_m^{(i)}-1},\tag{3.33}$$

其中 $$\theta_m^{(i)}=\frac{T_{m-1}^{(i)}-T_{m-2}^{(i+1)}}{T_{m-1}^{(i+1)}-T_{m-2}^{(i+1)}}.$$

先用(3.29)或(3.30)、(3.32)($1\leqslant m\leqslant l$), 再用(3.31)或(3.33)计算 $T_m^{(i)}$, 从而推算极限值 $T(0)$ 的方法, 称为 **Larkin 算法**.

在实际应用中, 常常采用(3.32)算 $T_1^{(i)}$, 用(3.33)算 $T_m^{(i)}$ ($m>1$). 这相当于 Larkin 的 B_1 算法. 此时所用插值有理函数的次数顺序为(据(3.17′))

$$0/0,\ 0/1,\ 1/1,\ 1/2,\ 2/2,\ 2/3,\ 3/3,\ \cdots.$$

可见分母次数跟分子次数相同或高一次. 若令

$$T_{-1}^{(i)}=0, \tag{3.34}$$

计算 $T_1^{(i)}$ 的(3.32)还可统一写为(3.33). 利用(3.34)、(3.33)推算 $T(0)$ 近似值 $T_m^{(i)}$ 的方法称为 **Stoer 有理式外推算法**.

在用 Stoer 外推算法编写计算机程序时, 为了避免在计算过程中反复相减损失有效数字, 可令

$$\Delta T_m^{(i)}=T_m^{(i)}-T_{m-1}^{(i+1)},\quad C_m^{(i)}=T_m^{(i)}-T_{m-1}^{(i)},$$
$$W_m^{(i)}=T_m^{(i)}-T_m^{(i-1)}=C_m^{(i)}-\Delta T_m^{(i-1)}.$$

注意 $T_{-1}^{(i)}=0$, $T_0^{(i)}=T(h_i)$, Stoer 算法的计算公式可改写为

$$\begin{cases}\Delta T_0^{(i)}=T(h_i),\quad C_0^{(i)}=T(h_i),\\ W_0^{(i)}=C_0^{(i)}-\Delta T_0^{(i-1)},\\ \Delta T_m^{(i)}=\dfrac{C_{m-1}^{(i+1)}W_{m-1}^{(i+1)}}{\dfrac{h_i}{h_{i+m}}\Delta T_{m-1}^{(i)}-C_{m-1}^{(i+1)}},\\ C_m^{(i)}=W_{m-1}^{(i+1)}+\Delta T_m^{(i)},\quad W_m^{(i)}=C_m^{(i)}-\Delta T_m^{(i-1)},\\ T_m^{(i)}=\sum\limits_{j=0}^{m}\Delta T_{m-j}^{(i+j)},\ m=1,\ 2,\ \cdots.\end{cases} \tag{3.35}$$

根据(3.35)推算 $T(0)$ 的方法称为 **Bulirsch-Stoer 有理式外推算法**. 具体编写计算机程序时, 可用一个一维数组存放 $\Delta T_m^{(i)}$, 而 $C_m^{(i)}$ 和 $W_m^{(i)}$ 只需用简单变量表示.

在实际应用中，还常常用 Larkin A_1 算法，来推算 $h\to\infty$ 时 $T(h)$ 的极限值. 根据(3.17)，A_1 算法产生的有理函数 $T_m^{(i)}(h)$ 的次数顺序为

$$0/0,\ 1/0,\ 1/1,\ 2/1,\ 2/2,\ 3/2,\ 3/3,\ \cdots.$$

可见 m 为偶数时，$T_m^{(i)}(h)$ 的分子、分母次数相同，$h\to\infty$ 时 $T_m^{(i)}(h)$ 有极限值；m 为奇数时，$T_m^{(i)}(h)$ 的分子比分母的次数高 1 次，$h\to\infty$ 时 $h/T_m^{(i)}(h)$ 趋于有限值. 我们把这些极限值仍记为 $T_m^{(i)}$. 令 $T_{-1}^{(i)}=0$，在公式(3.12)、(3.14)中令 $h\to\infty$，则知 $m>0$ 时

$$T_m^{(i)}=T_{m-2}^{(i+1)}+\frac{h_{i+m}-h_i}{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}. \tag{3.36}$$

根据 $T_{-1}^{(i)}=0$、$T_0^{(i)}=T(h_i)$ 及(3.36)推算 $T(\infty)$ 近似值 $T_m^{(i)}$(m 为偶数)的方法，称为 **ρ 算法或倒差商算法**. 它是 1956 年 Wynn 提出来的.

在(3.36)中，如果 $h_i=i+1$，则(3.36)变为

$$T_m^{(i)}=T_{m-2}^{(i+1)}+m/(T_{m-1}^{(i+1)}-T_{m-1}^{(i)}). \tag{3.37}$$

利用此式推算 $T(\infty)$ 近似值 $T_m^{(i)}$(m 为偶数)的方法，称为**带权的 ε 算法**. 这名称的来源，是它和 ε 算法很相似. ε 算法的递推公式为

$$T_m^{(i)}=T_{m-2}^{(i+1)}+1/(T_{m-1}^{(i+1)}-T_{m-1}^{(i)}). \tag{3.38}$$

ε 算法是推算 $T(\infty)$ 的有力工具，下一章将专门讨论. 比较(3.37)和(3.38)，自然使人想起，令

$$T_m^{(i)}=T_{m-2}^{(i+1)}+U_m^{(i)}/(T_{m-1}^{(i+1)}-T_{m-1}^{(i)}), \tag{3.39}$$

适当选取 $U_m^{(i)}$，有可能得到使 $\{T_m^{(i)}\}$ 收敛最快的外推算法. 这种想法是 1971 年 L. Wuytack 提出来的，不过至今还未见进一步的具体结果.

有理式外推法计算 $T_m^{(i)}$ 时往往要用到前二列的元素

$T_{m-2}^{(i+2)}$、$T_{m-1}^{(i+1)}$、$T_{m-1}^{(i)}$，所以 $T_m^{(i)}$ 常常排成表 5 的形式.

表 5　有理式外推表

	$T_0^{(0)}$				
$T_{-1}^{(1)}$		$T_1^{(0)}$			
	$T_0^{(1)}$		$T_2^{(0)}$		
$T_{-1}^{(2)}$		$T_1^{(1)}$		$T_3^{(0)}$	
	$T_0^{(2)}$		$T_2^{(1)}$		$T_4^{(0)}$
$T_{-1}^{(3)}$		$T_1^{(2)}$		$T_3^{(1)}$	
	$T_0^{(3)}$		$T_2^{(2)}$		⋮
$T_{-1}^{(4)}$		$T_1^{(3)}$		⋮	
	$T_0^{(4)}$		⋮		
⋮		⋮			

§5 误差估计

设 $T(h)$ 具有渐近展开式

$$T(h)=\tau_0+\tau_1 h+\tau_2 h^2+\cdots+\tau_N h^N+\tau_{N+1}(h)h^{N+1}. \tag{3.40}$$

$\tau_{N+1}(h)=\tau_{N+1}+o(h)$. 又设 A_m^i 是具有如下性质的线性算子:

$$\begin{cases}(\mathrm{a})\quad A_m^i T(h)=\sum_{k=i}^{i+m} C_{mk}^{(i)} T(h_k);\\(\mathrm{b})\quad A_m^i 1=1;\\(\mathrm{c})\quad A_m^i h^j=0\quad (j=1,\ 2,\ \cdots,\ m).\end{cases} \tag{3.41}$$

由(2.52)知

$$C_{mk}^{(i)}=\prod_{j=i}^{i+m}{}' \frac{h_j}{h_j-h_k}.$$

由于 $T_m^{(i)}(h)=R_{\mu\nu}^{(i)}(h)=P_\mu^{(i)}(h)/Q_\nu^{(i)}(h)\,(m=\mu+\nu)$ 满足插值条件(3.1)，我们有

$$P_\mu^{(i)}(h_k)=T(h_k)Q_\nu^{(i)}(h_k),\quad k=i,\ i+1,\ \cdots,\ i+m.$$

两边同乘 $C_{mk}^{(i)}$ 再对 k 求和，则当 $\mu\leqslant N$ 时(即 $m=\mu+\nu\leqslant N$

$+\nu$ 时)有

$$
\begin{aligned}
a_0^{(i)} &= P_\mu^{(i)}(0) = \Lambda_m^i P_\mu^{(i)}(h) = \sum_{k=i}^{i+m} C_{mk}^{(i)} P_\mu^{(i)}(h_k) \\
&= \sum_{k=i}^{i+m} C_{mk}^{(i)} T(h_k) Q_\nu^{(i)}(h_k) = \Lambda_m^i \{T(h) Q_\nu^{(i)}(h)\} \\
&= T(0) b_0^{(i)} + \Lambda_m^i \{(\tau_{\mu+1} + o(h)) h^{\mu+1} Q_\nu^{(i)}(h)\} \\
&\approx T(0) b_0^{(i)} + \Lambda_m^i \{\tau_{\mu+1} h^{\mu+1} Q_\nu^{(i)}(h)\} \\
&= T(0) b_0^{(i)} + \tau_{\mu+1} b_\nu^{(i)} \Lambda_m^i h^{m+1} \\
&= T(0) b_0^{(i)} + \tau_{\mu+1} b_\nu^{(i)} \sum_{k=i}^{i+m} C_{mk}^{(i)} h_k^{m+1}.
\end{aligned} \tag{3.42}
$$

故

$$
\begin{aligned}
T_m^{(i)}(0) - T(0) &= \frac{a_0^{(i)}}{b_0^{(i)}} - T(0) \\
&\approx \tau_{\mu+1} \frac{b_\nu^{(i)}}{b_0^{(i)}} \sum_{k=i}^{i+m} C_{mk}^{(i)} h_k^{m+1}.
\end{aligned} \tag{3.43}
$$

把 h^{m+1} 看作 $f(h)$, 按 Lagrange 插值公式(2.5)和(2.11)可知

$$
\begin{aligned}
\sum_{k=i}^{i+m} C_{mk}^{(i)} h_k^{m+1} &= \sum_{k=i}^{i+m} C_{mk}^{(i)} h_k^{m+1} - f(0) \\
&= -\frac{f^{(m+1)}(\xi)}{(m+1)!} \prod_{k=i}^{i+m} (o - h_k) \\
&= (-1)^m h_i h_{i+1} \cdots h_{i+m},
\end{aligned}
$$

故代入(3.43)得误差表达式

$$
T_m^{(i)}(0) - T(0) \approx (-1)^m \tau_{\mu+1} \frac{b_\nu^{(i)}}{b_0^{(i)}} h_i h_{i+1} \cdots h_{i+m}. \tag{3.44}
$$

这里的 $b_\nu^{(i)}/b_0^{(i)}$ 实质上跟 i 无关. 证明如下: 由(3.4)知

$$
b_0^{(i)} = \begin{vmatrix}
1 & h_i \cdots h_i^\mu & T(h_i) h_i & \cdots & T(h_i) h_i^\nu \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
1 & h_{i+m} \cdots h_{i+m}^\mu & T(h_{i+m}) h_{i+m} & \cdots & T(h_{i+m}) h_{i+m}^\nu
\end{vmatrix}
$$

$$
= \begin{vmatrix}
1 & h_i \cdots h_i^\mu & \tau_0 h_i + \tau_1 h_i^2 + \cdots & \cdots & \tau_0 h_i^\nu + \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
1 & h_{i+m} \cdots h_{i+m}^\mu & \tau_0 h_{i+m} + \tau_1 h_{i+m}^2 + \cdots & \cdots & \tau_0 h_{i+m}^\nu + \cdots
\end{vmatrix}
$$

从这个行列式的第 $\mu+2$ 列减去第 2 列的 τ_0 倍、第 3 列的 τ_1 倍、……、第 $\mu+1$ 列的 $\tau_{\mu-1}$ 倍；从第 $\mu+3$ 列减去第 3 列的 τ_0 倍、第 4 列的 τ_1 倍、……、第 $\mu+1$ 列的 $\tau_{\mu-2}$ 倍；……从最后一列减去第 $\nu+1$ 列的 τ_0 倍，第 $\nu+2$ 列的 τ_1 倍、……、第 $\mu+1$ 列的 $\tau_{\mu-\nu}$ 倍，则知

$$b_0^{(i)}=$$

$$\begin{vmatrix} 1 & h_i\cdots h_i^{\mu} & \tau_\mu h_i^{\mu+1}+\tau_{\mu+1}h_i^{\mu+2}+\cdots & \cdots & \tau_{\mu-\nu+1}h_i^{\mu+1}+\cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & h_{i+m}\cdots h_{i+m}^{\mu} & \tau_\mu h_{i+m}^{\mu+1}+\tau_{\mu+1}h_{i+m}^{\mu+2}+\cdots & \cdots & \tau_{\mu-\nu+1}h_{i+m}^{\mu+1}+\cdots \end{vmatrix}.$$

再对所得行列式的第 $\mu+1$ 列以后各列，从第 $\mu+2$ 列以外各列减去第 $\mu+2$ 列的适当倍数，消去这些列中含 $h^{\mu+1}$ 的项；从第 $\mu+3$ 列以外各列减去第 $\mu+3$ 列的适当倍数，消去这些列中含 $h^{\mu+2}$ 的项；……；从最后一列以外各列减去最后一列的适当倍数，消去这些列中含 h^m 的项．结果得到

$$b_0^{(i)}=$$

$$\begin{vmatrix} 1 & h_i\cdots h_i^{\mu} & \tau'_\mu h_i^{\mu+1}+o(h_i^{m+1}) & \cdots & \tau'_{m-1}h_i^{m}+o(h_i^{m}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & h_{i+m}\cdots h_{i+m}^{\mu} & \tau'_\mu h_{i+m}^{\mu+1}+o(h_{i+m}^{m+1}) & \cdots & \tau'_{m-1}h_{i+m}^{m}+o(h_{i+m}^{m+1}) \end{vmatrix}.$$

这里 τ'_μ、$\tau'_{\mu+1}$、…、τ'_{m-1} 只跟 τ_μ、$\tau_{\mu-1}$、…、$\tau_{\mu-\nu+1}$ 有关，而跟 h_i、h_{i+1}、…、h_{i+m} 无关．将它们提出行列式，略去高阶无穷小量，得到

$$b_0^{(i)}\approx\tau'_\mu\tau'_{\mu+1}\cdots\tau'_{m-1}\begin{vmatrix} 1 & h_i & \cdots & h_i^{m} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & h_{i+m} & \cdots & h_{i+m}^{m} \end{vmatrix}.$$

同理可得

$$b_\nu^{(i)}=\begin{vmatrix} 1 & h_i\cdots h_i^\mu & T(h_i) & T(h_i)h_i & \cdots & T(h_i)h_i^{\nu-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & h_{i+m}\cdots h_{i+m}^\mu & T(h_{i+m}) & T(h_{i+m})h_{i+m} & \cdots & T(h_{i+m})h_{i+m}^{\nu-1} \end{vmatrix}$$

$$\approx \tau''_\mu \tau''_{\mu+1}\cdots\tau''_{m-1}\begin{vmatrix} 1 & h_i & \cdots & h_i^m \\ 1 & h_{i+m} & \cdots & h_{i+m}^m \end{vmatrix},$$

其中 τ''_μ、$\tau''_{\mu+1}$、$\cdots$、τ''_{m-1} 也只跟 $\tau_{\mu+1}$、τ_μ、$\cdots$、$\tau_{\mu-\nu+2}$ 有关，而跟 h_i、h_{i+1}、$\cdots$、h_{i+m} 无关．于是

$$\frac{b_\nu^{(i)}}{b_0^{(i)}}\tau_{\mu+1}\approx\frac{\tau''_\mu\tau''_{\mu+1}\cdots\tau''_{m-1}}{\tau'_\mu\tau'_{\mu+1}\cdots\tau'_{m-1}}\tau_{\mu+1}=\tilde{\tau}_{\mu+1}.$$

这里 $\tilde{\tau}_{\mu+1}$ 只跟 $\tau_{\mu+1}$、τ_μ、$\cdots$、$\tau_{\mu-\nu+1}$ 有关，而跟 h_i、h_{i+1}、$\cdots$、h_{i+m} 无关．代入(3.44)，得到误差估计式

$$T_m^{(i)}(0)-T(0)\approx(-1)^m h_i h_{i+1}\cdots h_{i+m}\tilde{\tau}_{\mu+1}. \tag{3.45}$$

这里(3.45)类似于多项式外推法的误差估计式(2.47)．因此类似于(2.57)～(2.65)的推导，可得出 $T(0)$ 的渐近上下限与误差估计．注意这里 $r=1$，公式(2.14)不能用．现将结果罗列如下：设 $T_m^{(i)}=T_m^{(i)}(0)$，令

$$U_m^{(i)}\equiv(1+\alpha)T_m^{(i+1)}-\alpha T_m^{(i)}, \tag{3.46}$$

其中

$$\alpha=1+2/(h_i/h_{i+m}-1), \tag{3.47}$$

则 $T_m^{(i)}$、$U_m^{(i)}$ 是 $T(0)$ 的渐近上下限，从而当 i 充分大时

$$\left|\frac{1}{2}(T_m^{(i)}+U_m^{(i)})-T(0)\right|\leqslant\frac{1}{2}\left|T_m^{(i)}-U_m^{(i)}\right|. \tag{3.48}$$

如果简单地令

$$U_m^{(i)}\equiv 2T_m^{(i+1)}-T_m^{(i)}, \tag{3.49}$$

则当

$$2h_{i+m}/h_i<1 \tag{3.50}$$

时 $T_m^{(i)}$、$U_m^{(i)}$ 仍为 $T(0)$ 的渐近上下限，(3.48)仍成立，而且还有

$$|T_m^{(i+1)}-T(0)|\leqslant|T_m^{(i+1)}-T_m^{(i)}|, \tag{3.51}$$

$$|T_m^{(i+1)}-T(0)|\approx\frac{|T_m^{(i+1)}-T^{(i)}|}{h_i/h_{i+m}-1}. \tag{3.52}$$

而 i 充分大的标志是 $T_m^{(i)}$ 单调，或者

$$D_m^{(i)}\equiv\frac{h_i}{h_{i+m}}\cdot\frac{h_{i+m}-h_{i-1}}{h_{i+m+1}-h_i}\cdot\frac{T_m^{(i+1)}-T_m^{(i)}}{T_m^{(i)}-T_m^{(i-1)}}\approx 1. \tag{3.53}$$

$T(h)$ 和 $T_m^{(i)}(h)$ 当 $h\to\infty$ 时的极限值，可看作经变量替换 $\bar{h}=1/h$ 之后 $\bar{h}=0$ 时的值．由于我们用插值有理分式 $T_m^{(i)}(h)$ 推算 $T(\infty)$ 的值时，$T_m^{(i)}(h)$ 分子分母的次数相同，可见变量替换后

$$T_m^{(i)}\left(\frac{1}{\bar{h}}\right)=\frac{P_\mu^{(i)}(1/\bar{h})}{Q_\nu^{(i)}(1/\bar{h})}=\frac{\bar{h}^\mu P_\mu^{(i)}(1/\bar{h})}{\bar{h}^\mu Q_\nu^{(i)}(1/\bar{h})},$$

即 $T_m^{(i)}(1/\bar{h})$ 仍为 $\bar{h}$ 的有理分式，且满足插值条件

$$T_m^{(i)}(1/\bar{h}_k)=T(h_k)=T(1/\bar{h}_k)\quad(k=i,\ i+1,\ \cdots,\ i+m).$$

这样，如果 $T(h)$ 具有渐近展开式

$$\begin{aligned}T(h)&=\tau_0+\tau_1h^{-1}+\tau_2h^{-2}+\cdots+\tau_Nh^{-N}+\tau_{N+1}(h)h^{-N-1}\\&=\tau_0+\tau_1\bar{h}+\tau_2\bar{h}^2+\cdots+\tau_N\bar{h}^N+\tau_{N+1}(1/\bar{h})\bar{h}^{N+1},\end{aligned}$$

则前面关于 $T_m^{(i)}(0)$ 的误差估计式仍然成立，只是要把其中的 h 改为 $\bar{h}$ 即可．于是，如果仍令 $T_m^{(i)}$ 表示 $T_m^{(i)}(\infty)$，则当 m 为偶数且 $\mu\leqslant N$ 时有误差估计式

$$T_m^{(i)}-\tau_0\approx(-1)^m h_i^{-1}h_{i+1}^{-1}\cdots h_{i+m}^{-1}\tau_{\mu+1}. \tag{3.45'}$$

另外，将(3.47)、(3.50)、(3.52)、(3.53)中的 h 改为 h^{-1}，则(3.46)～(3.53)的全部结论也成立．

§6 收敛性与稳定性

我们在导出(3.42)时已经见到，

$$a_0^{(i)}=\sum_{k=i}^{i+m}C_{mk}^{(i)}T(h_k)Q_\nu^{(i)}(h_k).$$

由此，

$$T_m^{(i)}(0)=\frac{a_0^{(i)}}{b_0^{(i)}}=\sum_{k=i}^{i+m}C_{mk}^{(i)}\frac{Q_\nu^{(i)}(h_k)}{b_0^{(i)}}T(h_k)$$

$$=\sum_{k=i}^{i+m}\bar{C}_{mk}^{(i)}T(h_k). \tag{3.54}$$

其中 $\bar{C}_{mk}^{(i)}=C_{mk}^{(i)}Q_\nu^{(i)}(h_k)/b_0^{(i)}$. 这公式类似于(2.68)，而且

$$\sum_{k=i}^{i+m}\bar{C}_{mk}^{(i)}=\sum_{k=i}^{i+m}C_{mk}^{(i)}\frac{Q_\nu^{(i)}(h_k)}{b_0^{(i)}}=\Delta_m^i\left\{\frac{Q_\nu^{(i)}(h)}{b_0^{(i)}}\right\}=1,$$

说明类似于(2.69)的公式也成立. 显然，如果假定

$$|Q_\nu^{(i)}(h_k)/b_0^{(i)}|\leqslant c_i.$$

且 $\{h_i\}$ 满足第 2 章 § 6 的假定，即 $h_{i+1}/h_i\leqslant b<1$，则

$$\sum_{k=i}^{i+m}|\bar{C}_{mk}^{(i)}|\leqslant C_i\sum_{k=i}^{i+m}|C_{mk}|\leqslant MC_i=\bar{M},$$

说明类似于(2.70)的公式又成立. 此时，根据第 2 章 § 6 的证明，对固定的 k，

$$\lim_{m\to\infty}C_{m,i+k}^{(i)}=0,$$

故还有

$$\lim_{m\to\infty}\bar{C}_{m,i+k}^{(i)}=\lim_{m\to\infty}C_{m,i+k}^{(i)}Q_\nu^{(i)}(h_{i+k})/b_0^{(i)}=0.$$

这样，第 2 章 § 6 外推表对角线与列上元素收敛的全部依据，都是成立的. 于是，对于有理式外推法，若用 $T_m^{(i)}$ 推算 $T(0)=\tau_0$，则在(3.55)和 $h_{i+1}/h_i\leqslant b<1$ 的假定下，对角线和列都是收敛的，即对固定的 i 有

$$\lim_{m\to\infty}T_m^{(i)}=\tau_0,$$

而对固定的 m 有

$$\lim_{i\to\infty}T_m^{(i)}=\tau_0.$$

若用 $T_m^{(i)}$ 推算 $T(\infty)=\tau_0$，利用变量替换 $\bar{h}=1/h$ 可知，在

$$|\bar{Q}_\nu^{(i)}(h_k)/b_\nu^{(i)}|\leqslant c_i,\quad h_i/h_{i+1}\leqslant b<1$$

的假定下，上述结论仍成立。

关于有理式外推法的稳定性问题，跟第2章§6一样讨论，可知 b 越接近于1算法的稳定性越差。

最后应当指出，通过变量替换 $h=\bar{h}^r$，本节所有的结论在将 h 改为 $\bar{h}^r$ 时也成立。注意此时误差估计式(3.45)变为

$$T_m^{(i)}(0)-T(0)\approx(-1)^m\tilde{\tau}_{\mu+1}\bar{h}_i^r\bar{h}_{i+1}^r\cdots\bar{h}_{i+m}^r.$$

比较多项式外推算法的误差估计式(2.47)，可见两者几乎完全相同，只是 τ_{m+1} 与 $\tilde{\tau}_{\mu+1}$ 的不同。这说明，用有理式外推法算出的 $T_m^{(i)}$，跟用多项式外推法算出的 $T_m^{(i)}$，两者误差阶数完全相同，只是系数 $\tilde{\tau}_{\mu+1}$ 与 τ_{m+1} 不同。在很多情况下，$|\tilde{\tau}_{\mu+1}|$ 往往比 $|\tau_{m+1}|$ 小得多，这时采用有理式外推法，所得 $T_m^{(i)}$ 的误差就较小。

第 4 章

ε 算法

§1 ε算法的导出

多项式外推算法与有理式外推算法，实质上是在离散化近似值 $T(h)$ 具有渐近展开式

$$T(h)=\tau_0+\tau_1 h^r+\tau_2 h^{2r}+\cdots+\tau_N h^{Nr}+\tau_{N+1}(h)h^{(N+1)r}$$

的情况下，用插值多项式

$$P_m^{(i)}(h)=a_0+a_1 h^r+a_2 h^{2r}+\cdots+a_m h^{mr}$$

或插值有理分式

$$R_{\mu\nu}^{(i)}(h)=\frac{a_0+a_1 h^r+\cdots+a_\mu h^{\mu r}}{b_0+b_1 h^r+\cdots+a_\nu h^{\nu r}}$$

来近似替代 $T(h)$，用 $P_m^{(i)}(0)$ 或 $R_{\mu\nu}^{(i)}(0)$ 来推算 $T(0)$，或用 $R_{\mu\nu}^{(i)}(\infty)$ 来推算 $T(\infty)$. 由此自然想到，如果离散化参数改用 k 表示，而 τ_0 的离散化近似值 $T_k=T(k)$ 具有渐近展开式

$$T_k=\tau_0+\tau_1\lambda_1^k+\tau_2\lambda_2^k+\cdots+\tau_N\lambda_N^k+o(\lambda_N^k)$$

$$(k=0,\ 1,\ \cdots), \qquad (4.1)$$

其中常数 τ 和 λ 的值通常不知道，但知道

$$1>|\lambda_1|>|\lambda_2|>\cdots>|\lambda_N|>0,$$

那么自然应当用“插值多项式”

$$P_m(k)=a_0+a_1\lambda_1^k+a_2\lambda_2^k+\cdots+a_m\lambda_m^k \qquad (4.2)$$

来近似替代 T_k，用 $k\to\infty$ 时 $P_m(k)$ 的极限值 $P_m(\infty)=a_0$ 来推算 T_k 的极限值 τ_0. 这里 $P_m(k)$ 满足插值条件

$$P_m(k)=a_0+a_1\lambda_1^k+a_2\lambda_2^k+\cdots+a_m\lambda_m^k=T_k$$
$$(k=i,\ i+1,\ \cdots,\ i+2m). \qquad (4.3)$$

即满足方程组

$$\left.\begin{aligned}
&a_0+a_1\lambda_1^i+a_2\lambda_2^i+\cdots+a_m\lambda_m^i=T_i,\\
&a_0+a_1\lambda_1^{i+1}+a_2\lambda_2^{i+1}+\cdots+a_m\lambda_m^{i+1}=T_{i+1},\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&a_0+a_1\lambda_1^{i+2m}+a_2\lambda_2^{i+2m}+\cdots+a_m\lambda_m^{i+2m}=T_{i+2m}.
\end{aligned}\right\} \qquad (4.4)$$

显然，要使这种想法切实可行，首先必须找到计算 a_0 的简单算法.

设 λ_1、λ_2、$\cdots$、λ_m 是如下 m 次代数方程的根:

$$c_0+c_1x+c_2x^2+\cdots+c_mx^m=0. \qquad (4.5)$$

将(4.4)的前 $m+1$ 方程分别乘 c_0, c_1, $\cdots$, c_m, 然后相加，并考虑到(4.5), 则得

$$c_0(a_0-T_i)+c_1(a_0-T_{i+1})+\cdots+c_m(a_0-T_{i+m})=0.$$

将(4.4)的 $1\sim m+1$ 个方程、$2\sim m+2$ 个方程、……、$m\sim 2m$ 个方程照此处理，得到

$$\left.\begin{aligned}
&c_0(a_0-T_i)+c_1(a_0-T_{i+1})+\cdots+c_m(a_0-T_{i+m})=0,\\
&c_0(a_0-T_{i+1})+c_1(a_0-T_{i+2})+\cdots+c_m(a_0-T_{i+m+1})=0,\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&c_0(a_0-T_{i+m})+c_1(a_0+T_{i+m+1})+\cdots+c_m(a_0-T_{i+2m})=0.
\end{aligned}\right\}$$
$$(4.6)$$

这是 $m+1$ 个系数 c_0、c_1、$\cdots$、c_m 的线性齐次方程组，它有不全为 0 的解，故

$$\begin{vmatrix}
a_0-T_i & a_0-T_{i+1} & \cdots & a_0-T_{i+m}\\
a_0-T_{i+1} & a_0-T_{i+2} & \cdots & a_0-T_{i+m+1}\\
\cdots & \cdots & \cdots & \cdots\\
a_0-T_{i+m} & a_0-T_{i+m+1} & \cdots & a_0-T_{i+2m}
\end{vmatrix}=0.$$

从后面 m 行的每一行减去前一行，令 $\Delta T_j=T_{j+1}-T_j$, 得到

$$\begin{vmatrix} a_0-T_i & a_0-T_{i+1} & \cdots & a_0-T_{i+m} \\ \Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+m} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+m-1} & \Delta T_{i+m} & \cdots & \Delta T_{i+2m-1} \end{vmatrix} = 0.$$

将它按第一行展开，可得（假定下式分母不为 0）：

$$a_0 = \frac{\begin{vmatrix} T_i & T_{i+1} & \cdots & T_{i+m} \\ \Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+m} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+m-1} & \Delta T_{i+m} & \cdots & \Delta T_{i+2m-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+m} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+m-1} & \Delta T_{i+m} & \cdots & \Delta T_{i+2m-1} \end{vmatrix}}, \qquad (4.7)$$

这就是计算 a_0 的公式. 它跟 T_i, T_{i+1}, $\cdots$, T_{i+2m} 等 $2m+1$ 个数有关，记作 $\varepsilon_{2m}^{(i)}$ 或 $e_m(T_i)$，即

$$\varepsilon_{2m}^{(i)} = e_m(T_i) = a_0. \qquad (4.8)$$

由于它是“插值多项式”(4.2)的极限值，可以设想它比 T_i, T_{i+1}, $\cdots$, T_{i+2m} 能更精确地表示 $\{T_k\}$ 的极限值 τ_0，即加速 $\{T_k\}$ 的收敛. 利用它加速 $\{T_k\}$ 收敛的这种方法，称为 **Sanks** $\boldsymbol{e_m(T_i)}$变换.

当 $m=0$ 时，由(4.7)和(4.6)得

$$\varepsilon_0^{(i)} = T_i. \qquad (4.9)$$

当 $m=1$ 时，则得

$$\varepsilon_2^{(i)} = \begin{vmatrix} T_i & T_{i+1} \\ \Delta T_i & \Delta T_{i+1} \end{vmatrix} \Big/ \begin{vmatrix} 1 & 1 \\ \Delta T_i & \Delta T_{i+1} \end{vmatrix}$$
$$= \frac{T_i \Delta T_{i+1} - T_{i+1} \Delta T_i}{\Delta T_{i+1} - \Delta T_i},$$

或者

$$\varepsilon_2^{(i)}=\frac{T_{i+2}T_i-T_{i+1}^2}{T_{i+2}-2T_{i+1}+T_i}=T_{i+2}-\frac{(\Delta T_{i+1})^2}{\Delta^2 T_i}. \tag{4.10}$$

这就是著名的 **Aitken Δ^2 加速法.**

对于 $m\neq1$ 的情形,利用行列式计算 $\varepsilon_{2m}^{(i)}$ 是很麻烦的. P. Wynn 1956 年证明,如果令

$$\varepsilon_{2m+1}^{(i)}=1/e_m(\Delta T_i), \tag{4.11}$$

及 $\varepsilon_{-1}^{(i)}=0$, 则当 $m=0, 1, 2, \cdots$时有

$$\varepsilon_{m+1}^{(i)}=\varepsilon_{m-1}^{(i+1)}+(\varepsilon_m^{(i+1)}-\varepsilon_m^{(i)})^{-1}. \tag{4.12}$$

把 $\varepsilon_m^{(i)}$ 排成表 6, 按照这个递推关系, $\varepsilon_{2m}^{(i)}$ 就很容易计算了. 利用 $\varepsilon_{-1}^{(i)}=0$、(4.9)和(4.12)推算 τ_0 近似值 $\varepsilon_{2m}^{(i)}$ 的方法, 称为 **ε 算法.**

表6 ε表

$\varepsilon_{-1}^{(0)}=0$						
	$\varepsilon_0^{(0)}=T_0$					
$\varepsilon_{-1}^{(1)}=0$		$\varepsilon_1^{(0)}$				
	$\varepsilon_0^{(1)}=T_1$		$\varepsilon_2^{(0)}$			
$\varepsilon_{-1}^{(2)}=0$		$\varepsilon_1^{(1)}$		$\varepsilon_3^{(0)}$		
	$\varepsilon_0^{(2)}=T_2$		$\varepsilon_2^{(1)}$		$\varepsilon_4^{(0)}$	
$\varepsilon_{-1}^{(3)}=0$		$\varepsilon_1^{(2)}$		$\varepsilon_3^{(1)}$		$\cdots$
	$\varepsilon_0^{(3)}=T_3$		$\varepsilon_2^{(1)}$		$\varepsilon_4^{(1)}$	
$\varepsilon_{-1}^{(4)}=0$		$\varepsilon_1^{(3)}$		$\varepsilon_3^{(2)}$		$\cdots$
	$\varepsilon_0^{(4)}=T_4$		$\varepsilon_2^{(3)}$		$\varepsilon_4^{(2)}$	
$\varepsilon_{-1}^{(5)}=0$		$\varepsilon_1^{(4)}$		$\varepsilon_3^{(3)}$		$\cdots$
	$\varepsilon_0^{(5)}=T_5$		$\varepsilon_2^{(4)}$		$\vdots$	
$\varepsilon_{-1}^{(6)}=0$		$\varepsilon_1^{(5)}$		$\vdots$		
	$\varepsilon_0^{(6)}=T_6$		$\vdots$			
$\varepsilon_{-1}^{(7)}=0$		$\vdots$				
	$\vdots$					
$\vdots$						

为了证明递推关系(4.12), 需要引用行列式的这样一个性质: 如果恒等式的各项是行列式余子式的齐次式,则将所有余子式的对角线元素同时延长或缩短,所得等式仍为恒等式.

例1. 因为

$$\begin{vmatrix} h_1 & h_4 \\ d_1 & d_4 \end{vmatrix} |a_1| - \begin{vmatrix} a_1 & a_4 \\ d_1 & d_4 \end{vmatrix} |h_1|$$

$$= \begin{vmatrix} h_1 & h_4 \\ a_1 & a_4 \end{vmatrix} |d_1|, \tag{4.13}$$

这里$|a_1|$表示一阶行列式，即$|a_1| = a_1$，所以

$$\begin{vmatrix} h_1 & h_2 & h_3 & h_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{vmatrix} \begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} - \begin{vmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{vmatrix} \begin{vmatrix} h_1 & h_2 & h_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix}$$

$$= \begin{vmatrix} h_1 & h_2 & h_3 & h_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ a_1 & a_2 & a_3 & a_4 \end{vmatrix} \begin{vmatrix} d_1 & d_2 & d_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix}.$$

例2. 因为

$$|c_3||d_4| - |c_4||d_4| = \begin{vmatrix} c_3 & c_4 \\ d_3 & d_4 \end{vmatrix} \cdot 1. \tag{4.14}$$

这里1表示0阶行列式，所以

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} \begin{vmatrix} a_1 & a_2 & a_4 \\ b_1 & b_2 & b_4 \\ d_1 & d_2 & d_4 \end{vmatrix} - \begin{vmatrix} a_1 & a_2 & a_4 \\ b_1 & b_2 & b_4 \\ c_1 & c_2 & c_4 \end{vmatrix} \begin{vmatrix} a_1 & a_2 & a_4 \\ b_1 & b_2 & b_4 \\ d_1 & d_2 & d_4 \end{vmatrix}$$

$$= \begin{vmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{vmatrix} \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix}.$$

这一性质的证明可阅本章参考文献[3].

现在我们来证明(4.12),即证明

$$\varepsilon_{m+1}^{(i)}-\varepsilon_{m-1}^{(i+1)}=(\varepsilon_m^{(i+1)}-\varepsilon_m^{(i)})^{-1}. \qquad (4.15)$$

当 $m=0$ 时,

$$\begin{aligned}\text{左边}&=\varepsilon_1^{(i)}-\varepsilon_{-1}^{(i+1)}=1/e_0(\Delta T_i)-0\\&=1/\varepsilon_0^{(i)}(\Delta T_i)=1/\Delta T_i\\&=1/(T_{i+1}-T_i)=(\varepsilon_0^{(i+1)}-\varepsilon_0^{(i)})^{-1}\\&=\text{右边}.\end{aligned}$$

当 $m=2n$ 时,

$$\text{左边}=\varepsilon_{2n+1}^{(i)}-\varepsilon_{2n-1}^{(i+1)}=1/e_{2n}(\Delta T_i)-1/e_{2n-2}(\Delta T_{i+1})$$

$$=\frac{\begin{vmatrix}1 & 1 & \cdots & 1\\ \Delta^2T_i & \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-1}\end{vmatrix}}{\begin{vmatrix}\Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+n}\\ \Delta^2T_i & \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-1}\end{vmatrix}}$$

$$-\frac{\begin{vmatrix}1 & 1 & \cdots & 1\\ \Delta^2T_{i+1} & \Delta^2T_{i+2} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-2}\end{vmatrix}}{\begin{vmatrix}\Delta T_{i+1} & \Delta T_{i+2} & \cdots & \Delta T_{i+n}\\ \Delta^2T_{i+1} & \Delta^2T_{i+2} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-2}\end{vmatrix}}.$$

将第一个分式分子、分母行列式的第一列移到最后一列，则得

$$
左边=\frac{\begin{vmatrix}1 & \cdots & 1 & 1\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n} & \Delta^2T_i\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-1} & \Delta^2T_{i+n-1}\end{vmatrix}}{\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_i\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n} & \Delta^2T_i\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-1} & \Delta^2T_{i+n-1}\end{vmatrix}}
-\frac{\begin{vmatrix}1 & \cdots & 1\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \cdots & \Delta^2T_{i+2n-2}\end{vmatrix}}{\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n}\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \cdots & \Delta^2T_{i+2n-2}\end{vmatrix}}.
$$

将两分式通分，将分子和例 1 对比，知分子是(4.13)左边的延伸，故

$$
左边=\frac{\begin{vmatrix}1 & \cdots & 1 & 1\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n} & \Delta^2T_i\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \cdots & \Delta^2T_{i+2n-2} & \Delta^2T_{i+n-2}\\ \Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_i\end{vmatrix}\begin{vmatrix}\Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-1}\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \cdots & \Delta^2T_{i+2n-2}\end{vmatrix}}{\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_i\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n} & \Delta^2T_i\\ \cdots & \cdots & \cdots & \cdots\\ \Delta^2T_{i+n} & \cdots & \Delta^2T_{i+2n-1} & \Delta^2T_{i+n-1}\end{vmatrix}\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n}\\ \Delta^2T_{i+1} & \cdots & \Delta^2T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta^2T_{i+n-1} & \cdots & \Delta^2T_{i+2n-2}\end{vmatrix}}.
$$

将分子第一个行列式的最后一行移至第二行，第二个行列式

的第一行移至最后一行，将分母二行列式的第一行加到第二行，第二行加到第三行，……，则得

$$
\text{左边}=\frac{\begin{vmatrix} 1 & \cdots & 1 & 1 \\ \Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_{i} \\ \Delta^2 T_{i+1} & \cdots & \Delta^2 T_{i+n} & \Delta^2 T_{i} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta^2 T_{i+n-1} & \cdots & \Delta^2 T_{i+2n-2} & \Delta^2 T_{i+n-2} \end{vmatrix} \begin{vmatrix} \Delta^2 T_{i+1} & \cdots & \Delta^2 T_{i+n} \\ \cdots & \cdots & \cdots \\ \Delta^2 T_{i+n} & \cdots & \Delta^2 T_{i+2n-1} \end{vmatrix}}{\begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_{i} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n+1} & \cdots & \Delta T_{i+2n} & \Delta T_{i+n} \end{vmatrix} \begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} \end{vmatrix}}.
$$

将分子第一个行列式第二行加到第三行，第三行加到第四行，……；将分子第二个行列式改写为

$$
\begin{vmatrix} 1 & 0 & \cdots & 0 \\ \Delta T_{i+1} & \Delta^2 T_{i+1} & \cdots & \Delta^2 T_{i+n} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \Delta^2 T_{i+n} & \cdots & \Delta^2 T_{i+2n-1} \end{vmatrix},
$$

然后将第一列加到第二列，第二列加到第三列，……；最后将分子、分母的第一个行列式的最后一列移到第一列，则得

$$
\text{左边}=\frac{\begin{vmatrix} 1 & \cdots & 1 \\ \Delta T_{i} & \cdots & \Delta T_{i+n} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} \end{vmatrix} \begin{vmatrix} 1 & \cdots & 1 \\ \Delta T_{i+1} & \cdots & \Delta T_{i+n+1} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n} \end{vmatrix}}{\begin{vmatrix} \Delta T_{i} & \cdots & \Delta T_{i+n} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n} \end{vmatrix} \begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} \end{vmatrix}}.
$$

另一方面，(4.15)的

右边$=(\varepsilon_{2n}^{(i+1)}-\varepsilon_{2n}^{(i)})^{-1}=(e_n(T_{i+1})-e_n(T_i))^{-1}$

$$=\left\{\frac{\begin{vmatrix} T_{i+1} & T_{i+2} & \cdots & T_{i+n+1} \\ \Delta T_{i+1} & \Delta T_{i+2} & \cdots & \Delta T_{i+n+1} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \Delta T_{i+n+1} & \cdots & \Delta T_{i+2n} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta T_{i+1} & \Delta T_{i+2} & \cdots & \Delta T_{i+n+1} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \Delta T_{i+n+1} & \cdots & \Delta T_{i+2n} \end{vmatrix}} - \frac{\begin{vmatrix} T_i & T_{i+1} & \cdots & T_{i+n} \\ \Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+n} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n-1} & \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+n} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n-1} & \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} \end{vmatrix}}\right\}^{-1}$$

$$=\left\{\frac{\begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n+1} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n} \\ T_{i+1} & \cdots & T_{i+n+1} \end{vmatrix}}{\begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n+1} \\ \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n} \\ 1 & \cdots & 1 \end{vmatrix}} - \frac{\begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_i \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} & \Delta T_{i+n-1} \\ T_{i+1} & \cdots & T_{i+n} & T_i \end{vmatrix}}{\begin{vmatrix} \Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_i \\ \cdots & \cdots & \cdots & \cdots \\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} & \Delta T_{i+n-1} \\ 1 & \cdots & 1 & 1 \end{vmatrix}}\right\}^{-1},$$

将花括号内两分式通分，然后将分子跟例 2 对比，可见分子是(4.14)的延伸，故有

$$
\text{右边}=\left\{\frac{\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n+1} & \Delta T_i\\ \cdots & \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n} & \Delta T_{i+n-1}\\ T_{i+1} & \cdots & T_{i+n+1} & T_i\\ 1 & \cdots & 1 & 1\end{vmatrix}\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1}\end{vmatrix}}{\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n+1}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n}\\ 1 & \cdots & 1\end{vmatrix}\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n} & \Delta T_i\\ \cdots & \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1} & \Delta T_{i+n-1}\\ 1 & \cdots & 1 & 1\end{vmatrix}}\right\}^{-1}
$$

$$
=\frac{\begin{vmatrix}1 & \cdots & 1\\ \Delta T_i & \cdots & \Delta T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n-1} & \cdots & \Delta T_{i+2n-1}\end{vmatrix}\begin{vmatrix}1 & \cdots & 1\\ \Delta T_{i+1} & \cdots & \Delta T_{i+n+1}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n}\end{vmatrix}}{\begin{vmatrix}1 & 1 & \cdots & 1\\ T_i & T_{i+1} & \cdots & T_{i+n+1}\\ \Delta T_i & \Delta T_{i+1} & \cdots & \Delta T_{i+n+1}\\ \cdots & \cdots & \cdots & \cdots\\ \Delta T_{i+n-1} & \Delta T_{i+n} & \cdots & \Delta T_{i+2n}\end{vmatrix}\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1}\end{vmatrix}}
$$

$$
=\frac{\begin{vmatrix}1 & \cdots & 1\\ \Delta T_i & \cdots & \Delta T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n-1} & \cdots & \Delta T_{i+2n-1}\end{vmatrix}\begin{vmatrix}1 & \cdots & 1\\ \Delta T_{i+1} & \cdots & \Delta T_{i+n+1}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n}\end{vmatrix}}{\begin{vmatrix}\Delta T_i & \cdots & \Delta T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n}\end{vmatrix}\begin{vmatrix}\Delta T_{i+1} & \cdots & \Delta T_{i+n}\\ \cdots & \cdots & \cdots\\ \Delta T_{i+n} & \cdots & \Delta T_{i+2n-1}\end{vmatrix}}=\text{左边}.
$$

这就证明了(4.15)当 $m=2n$ 时成立. 同理可证 $m=2n+1$ 时也成立. 于是(4.15)和(4.12)当 $m=0, 1, 2, \cdots$ 时成立.

应当注意，我们在(4.1)中假定 $|\lambda_1|$, $|\lambda_2|$, $\cdots$, $|\lambda_n|$ 小于

1，只是为了保证 $k\to\infty$ 时 T_k 的极限值为 τ_0. 实际上在由(4.3)、(4.6)导出(4.7)和(4.12)时并未用到这个假定. 因而，ε 算法不仅可用来推算(4.1)中的 τ_0，而且还有许多其它的用处.

1962 年 P. Wynn 已把 ε 算法推广到向量和矩阵的情形. 对于(4.12)中用到向量 $\boldsymbol{x}=(x_1, x_2, \cdots, x_n)^T$ 的逆向量 $\boldsymbol{x}^{-1}$，他定义为

$$\boldsymbol{x}^{-1}=\bar{\boldsymbol{x}}\Big/\sum_{s=1}^{n} x_s\bar{x}_s.$$

其中 $\bar{\boldsymbol{x}}=(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)$，$\bar{x}_s$ 是 x_s 的共轭复数.

§2 ε 算法的若干性质

上面指出，由(4.3)或(4.6)可导出(4.7)和(4.12). 这说明，在(4.7)和(4.12)的分母不为 0 的情况下，ε 算法具有下述性质：

性质 1° 如果

$$T_k=a_0+\sum_{s=1}^{m} a_s\lambda_s^k,\ k=i,\ i+1,\ \cdots,\ i+2m,$$

a_s 为实数且 $a_s\neq 0$，$|\lambda_1|$、$|\lambda_2|$、$\cdots$、$|\lambda_m|$ 互不相同，那么

$$\varepsilon_{2m}^{(i)}=a_0.$$

性质 2° 如果对 $k=i,\ i+1,\ \cdots,\ i+m$ 有

$$\sum_{s=0}^{m} C_sT_{k+s}=a_0\sum_{s=0}^{m} C_s,\quad 即 \sum_{s=0}^{m} C_s(a_0-T_{k+s})=0,$$

$C_s\neq 0$，则当 $\sum\limits_{s=0}^{m} C_s\neq 0$ 时 $\varepsilon_{2m}^{(i)}=a_0$，否则 $\varepsilon_{2m}^{(i)}=0$.

这里 $\sum\limits_{s=0}^{m} C_s=0$ 的情形似乎未证过，其实不难得到：因为此时 C_0, C_1, $\cdots$, C_m 满足方程组

$$\sum_{s=0}^{m} C_s T_{k+s}=0,\ k=i,\ i+1,\ \cdots,\ i+m.$$

其系数行列式——即(4.7)右边的分子，应当为0，故 $a_0=0$.

由公式(4.7)可以推出 ε 算法的其它性质．例如有

$$e_m(a+bT_i)=\frac{\begin{vmatrix} a+bT_i & \cdots & a+bT_{i+m} \\ b\Delta T_i & \cdots & b\Delta T_{i+m} \\ \cdots & \cdots & \cdots \\ b\Delta T_{i+m-1} & \cdots & b\Delta T_{i+2m-1} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ b\Delta T_i & \cdots & b\Delta T_{i+m} \\ \cdots & \cdots & \cdots \\ b\Delta T_{i+m-1} & \cdots & b\Delta T_{i+2m-1} \end{vmatrix}}$$

$$=a+be_m(T_i),$$

$$e_m(\Delta\{a+bT_i\})=\frac{\begin{vmatrix} b\Delta T_i & \cdots & b\Delta T_{i+m} \\ b\Delta^2 T_i & \cdots & b\Delta^2 T_{i+m} \\ \cdots & \cdots & \cdots \\ b\Delta^2 T_{i+m-1} & \cdots & b\Delta^2 T_{i+2m-1} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ b\Delta^2 T_i & \cdots & b\Delta^2 T_{i+m} \\ \cdots & \cdots & \cdots \\ b\Delta^2 T_{i+m-1} & \cdots & b\Delta^2 T_{i+2m-1} \end{vmatrix}}$$

$$=be_m(\Delta T_i),$$

$$1/e_m(\Delta\{a+bT_i\})=b^{-1}/e_m(\Delta T_i).$$

这说明，ε 算法具有

性质 3° 设 ε 算法分别应用于 $\{T_k\}$ 与 $\{a+bT_k\}$ 时所得数组为 $\varepsilon_m^{(i)}$ 与 $\tilde{\varepsilon}_m^{(i)}$，则

$$\tilde{\varepsilon}_{2n}^{(i)}=a+b\varepsilon_{2n}^{(i)},\quad \tilde{\varepsilon}_{2n+1}^{(i)}=b^{-1}\varepsilon_{2n+1}^{(i)}. \tag{4.16}$$

设 $T_k=\sum_{s=0}^{k} a_s x^s$，由(4.7)还可知

$$e_m(T_i)=\frac{\begin{vmatrix} T_i & T_{i+1} & \cdots & T_{i+m} \\ a_{i+1}x^{i+1} & a_{i+2}x^{i+2} & \cdots & a_{i+m+1}x^{i+m+1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i+m}x^{i+m} & a_{i+m+1}x^{i+m+1} & \cdots & a_{i+2m}x^{i+2m} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ a_{i+1}x^{i+1} & \cdots & a_{i+m+1}x^{i+m+1} \\ \cdots & \cdots & \cdots \\ a_{i+m}x^{i+m} & \cdots & a_{i+2m}x^{i+2m} \end{vmatrix}}.$$

将分子、分母两个行列式的各列顺次乘 x^m、x^{m-1}、…、x^0，将各行顺次除以 1, x^{i+m+1}, …, x^{i+2m}，得

$$e_m(T_i)=\frac{\begin{vmatrix} T_ix^m & T_{i+1}x^{m-1} & \cdots & T_{i+m} \\ a_{i+1} & a_{i+2} & \cdots & a_{i+m+1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i+m} & a_{i+m+1} & \cdots & a_{i+2m} \end{vmatrix}}{\begin{vmatrix} x^m & x^{m-1} & \cdots & 1 \\ a_{i+1} & a_{i+2} & \cdots & a_{i+m+1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i+m} & a_{i+m+1} & \cdots & a_{i+2m} \end{vmatrix}}.$$

显然，它的分子是次数至多为 $i+m$ 的 x 多项式，分母是次数至多为m的 x 多项式. 将它们分别记为 $P_{i+m}(x)$、$Q_m(x)$，则

$$e_m(T_i)=P_{i+m}(x)/Q_m(x).$$

将 $P_{i+m}(x)$的第一行加上第二、三、……、$m+1$ 行的 x^{i+m+1}、x^{i+m+2}、…、x^{i+2m}倍，则 $P_{i+m}(x)$变为

$$P_{i+m}(x)=\begin{vmatrix} T_{i+m}x^m & T_{i+m+1}x^{m-1} & \cdots & T_{i+2m} \\ a_{i+1} & a_{i+2} & \cdots & a_{i+m+1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i+m} & a_{i+m+1} & \cdots & a_{i+2m} \end{vmatrix}.$$

于是

$$T_{i+2m}Q_m(x)-P_{i+m}(x)$$

$$=\begin{vmatrix} (T_{i+2m}-T_{i+m})x^m & (T_{i+2m}-T_{i+m+1})x^{m-1} & \cdots & T_{i+2m}-T_{i+2m} \\ a_{i+1} & a_{i+2} & \cdots & a_{i+m+1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i+m} & a_{i+m+1} & \cdots & a_{i+2m} \end{vmatrix}.$$

右边行列式是 x 的多项式，且次数至少为 $i+2m+1$．假定

$$f(x)=\sum_{s=0}^{\infty}a_s x^s$$

在 $x=0$ 的某个邻域内收敛，则

$$f(x)Q_m(x)-P_{i+m}(x)=[T_{i+2m}Q_m(x)-P_{i+m}(x)]$$

$$+Q_m(x)\sum_{s=i+2m+1}^{\infty}a_s x^s$$

的次数至少为 $i+2m+1$．这说明，$e_m(T_i)=P_{i+m}(x)/Q_m(x)$ 是 $f(x)$ 的 Padé 近似式——因为，所谓 $f(x)$ 的 Padé 近似式 $R_{\mu\nu}(x)=P_\mu(x)/Q_\nu(x)$，就是分子 $P_\mu(x)$ 至多为 μ 次、分母 $Q_\nu(x)$ 至多为 ν 次、而且满足如下条件的有理分式：

$$f(x)Q_\nu(x)-P_\mu(x)=(x^{\mu+\nu+1}),$$

其中 $(x^{\mu+\nu+1})$ 表示 x 次数至少为 $\mu+\nu+1$ 的级数．这样，ε 算法具有

性质 4° 设 $f(x)=\sum_{s=0}^{\infty}a_s x^s$ 在 $x=0$ 的某个邻域内收敛，$T_k=\sum_{s=0}^{k}a_s x^s$，则 $\varepsilon_{2m}^{(i)}=e_m(T_i)$ 是 $f(x)$ 的 $i+m/m$ 次 Padé 近似式．

以上性质都是从公式(4.7)推出来的．其实，直接由递推公式(4.12)出发，利用数学归纳法也可导出 ε 算法许多有用的性质，例如性质 3° 和下面的性质 5°．

性质 3° 的新证明：因为

$$\tilde{\varepsilon}_0^{(i)}=a+bT_i=a+b\varepsilon_0^{(i)},$$

$$\tilde{\varepsilon}_1^{(i)}=(\tilde{\varepsilon}_0^{(i+1)}-\tilde{\varepsilon}_0^{(i)})^{-1}=b^{-1}(\varepsilon_0^{(i+1)}-\varepsilon_0^{(i)})^{-1}=b^{-1}\varepsilon_1^{(i)},$$

说明(4.16)当 $n=0$ 时成立. 现设(4.16)直到 n 以前均成立, 则

$$\tilde{\varepsilon}_{2n+2}^{(i)}=a+b\varepsilon_{2n}^{(i+1)}+(b^{-1}\varepsilon_{2n+1}^{(i+1)}-b^{-1}\varepsilon_{2n+1}^{(i)})^{-1}=a+b\varepsilon_{2n+2}^{(i)},$$

$$\tilde{\varepsilon}_{2n+3}^{(i)}=b^{-1}\varepsilon_{2n+1}^{(i+1)}+(b\varepsilon_{2n+2}^{(i+1)}-b\varepsilon_{2n+2}^{(i)})^{-1}=b^{-1}\varepsilon_{2n+3}^{(i)}.$$

这证明(4.16)对任何 n 均成立.

性质 5° 设

$$T_k=\sum_{s=1}^{m}a_s\lambda_s^k,\ k=0,\ 1,\ 2,\ \cdots,\ a_s\neq 0,$$

$$\lambda_s\neq 1,\ |\lambda_1|>|\lambda_2|>\cdots>|\lambda_m|\geqslant 0,\ y_s=(\lambda_s-1)a_s,$$

则当 i 充分大时

$$\varepsilon_{2n}^{(i)}\sim\sum_{s=n+1}^{m}\lambda_s^{i+n}a_s,\qquad \varepsilon_{2n+1}^{(i)}\sim(\lambda_{n+1}^{i+n}y_{n+1})^{-1}. \tag{4.17}$$

事实上

$$\varepsilon_0^{(i)}=T_i=\sum_{s=1}^{m}a_s\lambda_s^i,$$

$$\varepsilon_1^{(i)}=(\varepsilon_0^{(i+1)}-\varepsilon_0^{(i)})^{-1}=\left\{\sum_{s=1}^{m}a_s(\lambda_s-1)\lambda_s^i\right\}^{-1}$$

$$=\left\{\lambda_1^i\left[y_1+\sum_{s=2}^{m}\left(\frac{\lambda_s}{\lambda_1}\right)^i y_s\right]\right\}^{-1}\sim(\lambda_1^i y_1)^{-1}.$$

这说明(4.17)当 $n=0$ 时成立. 现设(4.17)直到 n 以前均成立, 则

$$\varepsilon_{2n+2}^{(i)}\sim\sum_{s=n+1}^{m}\lambda_s^{i+1+n}a_s+[(\lambda_{n+1}^{i+1+n}y_{n+1})^{-1}-(\lambda_{n+1}^{i+n}y_{n+1})^{-1}]^{-1}$$

$$=\sum_{s=n+1}^{m}\lambda_s^{i+1+n}a_s+\frac{\lambda_{n+1}^{i+n+1}}{1-\lambda_{n+1}}y_{n+1}$$

$$=\sum_{s=n+1}^{m}\lambda_s^{i+n+1}a_s-\lambda_{n+1}^{i+n+1}a_{n+1}$$

$$=\sum_{s=n+2}^{m}\lambda_s^{i+n+1}a_s,$$

$$
\begin{aligned}
\varepsilon_{2n+3} &\sim \left(\lambda_{n+1}^{i+1+n} y_{n+1}\right)^{-1} + \left\{ \sum_{s=n+2}^{m} \lambda_s^{i+n+1} (\lambda_s - 1) a_s \right\}^{-1} \\
&= \left(\lambda_{n+1}^{i+1+n} y_{n+1}\right)^{-1} \\
&\quad + \left\{ \lambda_{n+2}^{i+n+1} \left[y_{n+2} + \sum_{s=n+3}^{m} \left(\frac{\lambda_s}{\lambda_{n+2}} \right)^{i+n+1} y_s \right] \right\}^{-1} \\
&\sim \left(\lambda_{n+1}^{i+1+n} y_{n+1}\right)^{-1} + \left(\lambda_{n+2}^{i+n+1} y_{n+2}\right)^{-1} \\
&= \left(\lambda_{n+2}^{i+n+1} y_{n+2}\right)^{-1} \left\{ 1 + \left(\frac{\lambda_{n+2}}{\lambda_{n+1}} \right)^{i+n+1} \frac{y_{n+2}}{y_{n+1}} \right\} \\
&\sim \left(\lambda_{n+2}^{i+n+1} y_{n+2}\right)^{-1}.
\end{aligned}
$$

这便证明了(4.17).

由(4.17)可知

$$
\varepsilon_{2n}^{(i)} \sim \lambda_{n+1}^{i+n} a_{n+1}, \quad \varepsilon_{2n+1}^{(i)} \sim \left(\lambda_{n+1}^{i+n} y_{n+1}\right)^{-1}. \tag{4.18}
$$

由此可见

$$
\varepsilon_{2n}^{(i+1)} \left(\varepsilon_{2n}^{(i)}\right)^{-1} \sim \lambda_{n+1}, \quad \varepsilon_{2n+1}^{(i+1)} \left(\varepsilon_{2n+1}^{(i)}\right)^{-1} \sim 1/\lambda_{n+1}. \tag{4.19}
$$

由性质 3° 和 5° 可知，如果

$$
T_k = a_0 + \sum_{s=1}^{m} a_s \lambda_s^k,
$$

则

$$
\varepsilon_{2n}^{(i)} - a_0 = e_n(T_i - a_0) \sim \sum_{s=n+1}^{m} \lambda_s^{i+n} a_s
$$

特别地有 $\qquad \varepsilon_{2m}^{(i)} - a_0 = 0, \quad$ 即 $\quad \varepsilon_{2m}^{(i)} = a_0.$

这就是性质 1°. 由性质 3° 和 5° 还可知，如果条件(4.1)成立，则有误差估计式

$$
\varepsilon_{2m}^{(i)} - \tau_0 = e_m(T_i - \tau_0) \sim \sum_{s=m+1}^{N+1} \lambda_s^{i+m} a_s.
$$

ε 算法的上述性质均可推广到向量的情形. 上面关于性质 3°、5°、1° 的证明对向量的情形也仍然有效.

性质 6° 如果

$$
T_k = a_0 + \sum_{s=1}^{p} a_s^{(k)} \lambda_s^k, \ k = 0, 1, 2, \cdots, \ a_s^{(k)} \neq 0
$$

同 k 有关，但具有周期性：

$$a_s^{(k)}=a_s^{(k+k_0)},\quad k_0 \text{ 为固定整数},$$

且 $1>|\lambda_1|>|\lambda_2|>\cdots>|\lambda_p|\geqslant 0$，那么

$$\varepsilon_{2k_0p}^{(0)}=a_0.$$

事实上，设

$$f(x)=T_0+(T_1-T_0)x+(T_2-T_1)x^2+\cdots,\quad \textbf{(4.20)}$$

则有

$$\begin{aligned}
f(x)&=T_0+x\sum_{s=1}^{p}(\lambda_s a_s^{(1)}-a_s^{(0)})\\
&\quad+x^2\sum_{s=1}^{p}(\lambda_s^2 a_s^{(2)}-\lambda_s a_s^{(1)})+\cdots\\
&=T_0+x\sum_{j=0}^{\infty}x^j\sum_{s=1}^{p}\lambda_s^j(\lambda_s a_s^{(j+1)}-a_s^{(j)})\\
&=T_0+x\sum_{s=1}^{p}\sum_{j=0}^{\infty}(\lambda_s x)^j(\lambda_s a_s^{(j+1)}-a_s^{(j)})\\
&=T_0+x\sum_{s=1}^{p}\sum_{l=0}^{\infty}(x\lambda_s)^{k_0 l}\\
&\quad\times\sum_{i=0}^{k_0-1}(x\lambda_s)^i(\lambda_s a_s^{(k_0 l+i+1)}-a_s^{(k_0 l+i)})\\
&=T_0+x\sum_{s=1}^{p}\left\{\sum_{i=0}^{k_0-1}(x\lambda_s)^i(\lambda_s a_s^{(i+1)}-a_s^{(i)})\right\}\Big/\\
&\quad[1-(x\lambda_s)^{k_0}]\\
&=a_0+\sum_{s=1}^{p}\{a_s^{(0)}[1-(x\lambda_s)^{k_0}]+(x\lambda_s a_s^{(1)}-xa_s^{(0)})\\
&\quad+x\lambda_s(x\lambda_s a_s^{(2)}-xa_s^{(1)}+\cdots+(x\lambda_s)^{k_0-2}\\
&\quad\times(x\lambda_s a_s^{(k_0-1)}-xa_s^{(k_0-2)})+(x\lambda_s)^{k_0-1}\\
&\quad\times(x\lambda_s a_s^{(0)}-xa_s^{(k_0-1)})\}/[1-(x\lambda_s)^{k_0}]\\
&=a_0+(1-x)\sum_{s=1}^{p}\left\{\sum_{i=0}^{k_0-1}a_s^{(i)}\lambda_s^i x^i\right\}\Big/[1-(x\lambda_s)^{k_0}].
\end{aligned}$$

这表明，$f(x)$是 k_0p/k_0p 次有理分式．而由(4.20)知

$$T_k=\left[T_0+\sum_{s=1}^{k}(T_s-T_{s-1})x^s\right]_{x=1},$$

故根据性质 4° 有

$$e_{2k_0p}^{(0)}=f(1)=a_0.$$

证毕．

第 5 章

外推法的应用(一)：数值积分与微分

§1 Euler-Maclaurin 求和公式

目前，外推法已应用到了数值分析的各类问题之中．这里首先介绍在数值积分中的应用．

在数值积分中，为了求积分

$$\tau_0 = If = \int_a^b f(x)dx \tag{5.1}$$

的值，常常使用如下形式的近似公式

$$If \approx Rf \equiv \sum_{k=1}^{n} w_k f(x_k), \tag{5.2}$$

也就是用被积函数 $f(x)$ 在 n 个节点 x_k 处的值的线性组合 Rf 来近似替代积分值 If．节点个数 n、节点 x_k 及其对应组合系数 w_k 的取法不同，便是不同的求积公式．例如，令 $h=(b-a)/n$, $x_k=a+kh$, $f_k=f(x_k)$, 则有

梯形求积公式

$$If \approx Tf \equiv h\left(\frac{1}{2}f_0+f_1+f_2+\cdots+f_{n-1}+\frac{1}{2}f_n\right), \tag{5.3}$$

广义矩形求积公式

$$If \approx R^{[n,\alpha]}f = h\sum_{k=1}^{n} f\left(a+\frac{h}{2}(2k-1+\alpha)\right), \quad |\alpha|\leqslant 1. \tag{5.4}$$

显然，按这些公式计算，h 不同所得积分近似值 Tf、$R^{[n,\alpha]}f$ 也不同，我们把它记为 $T(h)$．h 称为步长，是离散化

参数．可以设想，根据 If 的几个近似值 $T(h_0)$、$T(h_1)$、…，利用外推算法，有可能推算积分 If 的更精确近似值．但选择哪种算法才最有效呢？这就得研究 $T(h)$ 具有什么样的渐近展开式．

首先讨论梯形求积公式(5.3)．我们有如下展开式

$$Tf=If+\sum_{s=1}^{N}\frac{h^{2s}B_{2s}}{(2s)!}[f^{(2s-1)}(b)-f^{(2s-1)}(a)]+E_{N+1}, \tag{5.5}$$

$$\begin{aligned}E_{N+1}=&-\frac{h^{2N+3}}{(2N+2)!}\int_0^1[B_{2N+2}(t)-B_{2N+2}]\\&\times\sum_{k=0}^{N-1}f^{(2N+2)}(a+kh+th)\,dt\\&-\frac{h^{2N+2}}{(2N+2)!}(b-a)B_{2N+2}f^{(2N+2)}(\xi),\end{aligned} \tag{5.6}$$

其中 B_k 为 Bernoulli 常数，$B_k(t)$ 为 Bernoulli 多项式，被积函数 $f(x)$ 假定在 $[a, b]$ 上直到 $2N+2$ 阶导数连续，ξ 是 (a, b) 上某一点．这公式既可用来求积分，也可用来求级数的和，称为 **Euler-Maclaurin 求和公式**．

所谓 **Bernoulli 常数 B_k**，就是函数 $G(t)=t/(e^t-1)$ 的幂级数展开式的系数，即

$$G(t)=t/(e^t-1)=\sum_{k=0}^{\infty}\frac{B_k}{k!}t^k. \tag{5.7}$$

由于 $t=G(t)(e^t-1)$，故有

$$t=\left(B_0+\frac{B_1}{1!}t+\frac{B_2}{2!}t^2+\cdots\right)\left(\frac{1}{1!}t+\frac{1}{2!}t^2+\cdots\right).$$

比较两边 t^k 的系数，可见 $B_0=1$，而 $k>1$ 时

$$0=\frac{B_{k-1}}{(k-1)!1!}+\frac{B_{k-2}}{(k-2)!2!}+\cdots+\frac{B_0}{0!k!},$$

从而

$$B_k=\frac{k!}{k!0!}B_k+\frac{k!}{(k-1)!1!}B_{k-1}+$$
$$\frac{k!}{(k-2)!2!}B_{k-2}+\cdots+\frac{k!}{0!k!}B_0.$$

这样，如果把 B^k 理解为 B_k，则上式可改写为容易记忆的形式

$$B_k=(B+1)^k,\quad k>1. \tag{5.8}$$

这是计算 B_k 的递推关系．例如 $k=2, 3, 4, 5$ 时有

$$B_2=B_2+2B_1+1,$$
$$B_3=B_3+3B_2+3B_1+1,$$
$$B_4=B_4+4B_3+6B_2+4B_1+1,$$
$$B_5=B_5+5B_4+10B_3+10B_2+5B_1+1.$$

注意 $B_0=1$，由此顺次可得

$$B_1=-\frac{1}{2},$$
$$B_2=-\frac{1}{3}(3B_1+1)=\frac{1}{6},$$
$$B_3=-\frac{1}{4}(6B_2+4B_1+1)=0,$$
$$B_4=-\frac{1}{5}(10B_3+10B_2+5B_1+1)=-\frac{1}{30}.$$

注意$G(-t)=t+G(t)$，则有

$$\sum_{k=0}^{\infty}(-1)^k\frac{B_k}{k!}t^k=t+\sum_{k=0}^{\infty}\frac{B_k}{k!}t^k.$$

比较两边 t^{2k+1} 的系数，得$-B_{2k+1}=B_{2k+1}$，故

$$B_{2k+1}=0,\quad k>0.$$

正是由于除 $B_1=-\frac{1}{2}$ 外所有奇数号 Bernoulli 常数 B_{2k+1} 为0，许多人把 B_{2k} 记为 B_k.

所谓 Bernoulli 多项式 $B_k(x)$，就是函数

$$H(x,\ t)=te^{xt}/(e^t-1)$$

的幂级数展开式的系数，即

$$H(x,\ t)=\frac{te^{xt}}{e^t-1}=\sum_{k=0}^{\infty}\frac{B_k(x)}{k!}t^k. \tag{5.9}$$

显然 $H(x,\ t)=e^{xt}G(t)$，故有

$$\sum_{k=0}^{\infty}\frac{B_k(x)}{k!}t^k=\sum_{k=0}^{\infty}\frac{x^k}{k!}t^k\sum_{k=0}^{\infty}\frac{B_k}{k!}t^k.$$

比较两边 t^k 的系数可见

$$\frac{1}{k!}B_k(x)=\frac{x^k}{k!0!}B_0+\frac{x^{k-1}}{(k-1)!1!}B_1+\cdots+\frac{x^0}{0!k!}B_k,$$

所以

$$B_k(x)=(x+B)^k\quad(B^k=B_k). \tag{5.10}$$

这表明 $B_k(x)$ 是 k 次多项式，且由此可得

$$B_0(x)=1,\quad B_1(x)=x-\frac{1}{2},\quad B_2(x)=x^2-x+\frac{1}{6},\ \cdots.$$

由(5.10)还可得

$$B_k(0)=B^k=B_k. \tag{5.11}$$

根据 $B_k(x)$ 的定义式(5.9)还可导出 Bernoulli 多项式的许多重要性质．例如将(5.9)两边对 x 求导，得

$$\sum_{k=0}^{\infty}\frac{B_k'(x)}{k!}t^k=\frac{t^2e^{xt}}{e^t-1}=\sum_{k=0}^{\infty}\frac{B_k}{k!}t^{k+1}.$$

比较两边 t^k 的系数，得到 Bernoulli 多项式的求导公式

$$B_k'(x)=kB_{k-1}(x)\quad(k>0). \tag{5.12}$$

又由(5.9)有

$$\sum_{k=0}^{\infty}\frac{B_k(1-x)}{k!}t^k=\frac{te^{(1-x)t}}{e^t-1}=\frac{(-t)e^{x(-t)}}{e^{-t}-1}$$

$$=\sum_{k=0}^{\infty}\frac{B_k(x)}{k!}(-t)^k.$$

比较两边 t^k 的系数，得

$$B_k(1-x)=(-1)^k B_k(x). \qquad (5.13)$$

由此，$B_{2k}(1)=B_{2k}(0)$，而 $k>0$ 时

$$B_{2k+1}(1)=B_{2k+1}(0)=B_{2k+1}\left(\frac{1}{2}\right)=0. \qquad (5.14)$$

根据(5.12)和(5.13)可以证明，$B_{2k+1}(x)$在区间[0,1]上只有0，$\frac{1}{2}$和1这三个零点，而$y_{2k}(x)=B_{2k}(x)-B_{2k}$在区间[0, 1]上不变号.

事实上，假定$B_{2k+1}(x)$在[0, 1]上还另有零点，则按微分学的 Rolle 定理，$B'_{2k+1}(x)$在此零点所在的半个区间内有二零点，从而$B''_{2k+1}(x)=(2k+1)2kB_{2k-1}(x)$在这半个区间内有一零点．这说明，$B_{2k-1}(x)$除0，$\frac{1}{2}$和1外，还另有零点．依此类推，可知$B_3(x)$在[0, 1]上至少有四个零点，这同$B_3(x)$为三次多项式矛盾．这就证明了$B_{2k+1}(x)$在[0, 1]上只有0，$\frac{1}{2}$和1这三个零点．又若$y_{2k}(x)$在[0, 1]上变号，则由$y_{2k}(x)$的连续性知，$y_{2k}(x)$在(0, 1)上应有一零点．而由(5.11)及$B_{2k}(1)=B_{2k}(0)$知$y_{2k}(1)=y_{2k}(0)=0$，故根据 Rolle 定理知$y'_{2k}(x)=2kB_{2k-1}(x)$在(0, 1)上至少有两零点．这同$B_{2k-1}(x)$在[0, 1]上只有0，$\frac{1}{2}$和1这三零点的结论矛盾．因此$y_{2k}(x)$在[0, 1]上不变号．证毕.

根据(5.12)和(5.13)还可看出

$$B_k^{(l)}(x)=k(k-1)\cdots(k-l+1)B_{k-l}(x),$$

$$B_k^{(l)}(0)=\frac{k!}{(k-l)!}B_{k-l},$$

$$B_k^{(l)}(1)=(-1)^{k-l}\frac{k!}{(k-l)!}B_{n-k},$$

或

$$B_k^{(k-l)}(0)=\frac{k!}{l!}B_l,\quad B_k^{(k-l)}(1)=(-1)^l\frac{k!}{l!}B_l. \qquad (5.15)$$

根据 $B_k(x)$的求导公式(5.12)可算出 $B_k(x)$的 Fourie 系数

$$a_l^{(k)}=2\int_0^1 B_k(x)\cos 2\pi lx\,dx,$$

$$b_l^{(k)}=2\int_0^1 B_k(x)\sin 2\pi lx\,dx,$$

从而得出 $B_k(x)$的三角级数展开式

$$B_{2k}(x)=\frac{(-1)^{k-1}(2k)!}{2^{2k-1}\pi^{2k}}\sum_{l=1}^{\infty}\frac{\cos 2\pi lx}{l^{2k}},$$

$$B_{2k+1}(x)=\frac{(-1)^{k+1}(2k+2)!}{2^{2k+1}\pi^{2k+1}}\sum_{l=1}^{\infty}\frac{\sin 2\pi lx}{l^{2k+1}},$$

以及

$$B_{2k}=B_{2k}(0)=\frac{(-1)^k(2k)!}{2^{2k-1}\pi^{2k}}\sum_{l=1}^{\infty}\frac{1}{l^{2k}}$$

$$=\frac{(-1)^k(2k)!}{2^{2k-1}\pi^{2k}}\zeta(2k), \qquad (5.16)$$

其中 $\zeta(z)$是 Riemann 的 ζ 函数,

$$\zeta(z)=\sum_{l=1}^{\infty}\frac{1}{l^z}. \qquad (5.17)$$

为了证明 Euler-Maclaurin 求和公式，我们拟用如下著名的 **Darboux** 公式:

$$P_n^{(n)}(0)[f(x)-f(a)]$$

$$=\sum_{k=1}^{n}(-1)^{k-1}(x-a)^k\{P_n^{(n-k)}(1)f^{(k)}(x)$$

$$-P_n^{(n-k)}(0)f^{(k)}(a)\}+(-1)^n(x-a)^{n+1}$$

$$\times\int_0^1 P_n(t)f^{(n+1)}(a+t(x-a))dt, \qquad (5.18)$$

其中 $P_n(x)$是 n 次多项式，$f(x)$是 $n+1$ 阶导数连续的函数。这公式只要将下面的恒等式两边对 t 由 0 到 1 积分，并注意 $P_n^{(n)}(t)$是常数，便可得到：

$$
\begin{aligned}
&\frac{d}{dt}\sum_{k=1}^{n}(-1)^k(x-a)^kP_n^{(n-k)}(t)f^{(k)}(a+t(x-a))\\
&=-(x-a)P_n^{(n)}(t)f'(a+t(x-a))\\
&\quad-(x-a)^2P_n^{(n-1)}(t)f''(a+t(x-a))\\
&\quad+(x-a)^2P_n^{(n-1)}(t)f''(a+t(x-a))\\
&\quad+(x-a)^3P_n^{(n-2)}(t)f'''(a+t(x-a))\\
&\quad-\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&\quad+(-1)^n(x-a)^nP_n'(t)f^{(n)}(a+t(x-a))\\
&\quad+(-1)^n(x-a)^{n+1}P_n(t)f^{(n+1)}(a+t(x-a))\\
&=-(x-a)P_n^{(n)}(t)f'(a+t(x-a))\\
&\quad+(-1)^n(x-a)^{n+1}P_n(t)f^{(n+1)}(a+t(x-a)).
\end{aligned}
$$

顺便指出，在(5.18)中令 $P_n(x)=(x-1)^n$，便得 Taylor 公式。

现在我们来证明 Euler-Maclaurin 求和公式。令 $n=2N+2$，$P_n(t)=B_n(t)$，注意(5.15)，则由(5.18)得

$$
\begin{aligned}
&(2N+2)!\,[f(x)-f(a)]\\
&=\sum_{k=1}^{2N+2}(-1)^{k-1}(x-a)^k\frac{(2N+2)!}{k!}\\
&\quad\times\{(-1)^kf^{(k)}(x)-f^{(k)}(a)\}B_k\\
&\quad+(-1)^{2N+2}(x-a)^{2N+3}\\
&\quad\times\int_0^1B_{2N+2}(t)f^{(2N+3)}(a+t(x-a))dt.
\end{aligned}
$$

再注意 $B_1=-\frac{1}{2}$，$B_{2k+1}=0(k>0)$，得

$$
\begin{aligned}
f(x)-f(a)=&\frac{1}{2}[f'(x)+f'(a)]\\
&-\sum_{s=1}^{N}\frac{(x-a)^{2s}}{(2s)!}B_{2s}[f^{(2s)}(x)-f^{(2s)}(a)]\\
&+\frac{(x-a)^{2N+3}}{(2N+2)!}\int_0^1[B_{2N+2}(t)-B_{2N+2}]\\
&\times f^{(2N+3)}(a+t(x-a))dt.
\end{aligned}
$$

把 $f(x)$ 改为 $\int_a^x f(t)dt$, x 改为 $a+h$, 则上式变为

$$
\begin{aligned}
\frac{1}{2}[f(a)+f(a+h)]=&\int_a^{a+h}f(x)dx+\sum_{s=1}^{N}\frac{h^{2s}B_{2s}}{(2s)!}\\
&\times[f^{(2s-1)}(a+h)-f^{(2s-1)}(a)]\\
&-\frac{h^{2N+3}}{(2N+2)!}\int_0^1[B_{2N+2}(t)-B_{2N+2}]\\
&\times f^{(2N+2)}(a+th)dt.
\end{aligned}
$$

再将 a 改为 $a+h$, $a+2h$, $\cdots$, $a+(n-1)h$, $n=(b-a)/h$, 然后将所得式子相加,则得

$$
\begin{aligned}
Tf=&h\Big[\frac{1}{2}f(a)+f(a+h)+f(a+2h)+\cdots\\
&+f(a+(n-1)h)+\frac{1}{2}f(b)\Big]\\
=&\int_a^b f(x)dx+\sum_{s=1}^{N}\frac{h^{2s}B_{2s}}{(2s)!}[f^{(2s-1)}(b)-f^{(2s-1)}(a)]\\
&-\frac{h^{2N+3}}{(2N+2)!}\int_0^1[B_{2N+2}(t)-B_{2N+2}]\\
&\times\sum_{k=0}^{n-1}f^{(2N+2)}(a+kh+th)dt.
\end{aligned}
$$

这就是我们要证的 Euler–Maclaurin 求和公式(5.5),只是余项 E_{N+1} 的第二种表示式尚待证明.

其实这很容易证明:由于 $y_{2N+2}(x)=B_{2N+2}(x)-B_{2N+2}$ 在区间[0, 1]上不变号,故根据广义积分中值定理

$$\int_0^1 [B_{2N+2}(t)-B_{2N+2}]\sum_{k=0}^{n-1} f^{(2N+2)}(a+kh+th)dt$$

$$=\sum_{k=0}^{n-1} f^{(2N+2)}(a+kh+\theta_k h)\int_0^1 [B_{2N+2}(t)-B_{2N+2}]dt$$

$$=-B_{2N+2}\sum_{k=0}^{n-1} f^{(2N+2)}(a+kh+\theta_k h),$$

其中$0<\theta_k<1$. 而由于$f^{(2N+2)}(a+kh+th)$在$[0, 1]$上连续，必有

$$nm\leqslant\sum_{k=0}^{n-1} f^{(2N+2)}(a+kh+\theta_k h)\leqslant nM,$$

这里m和M分别表示$f^{(2N+2)}(x)$在$[a, b]$上的最小值与最大值，从而根据连续函数的介值定理，一定存在ξ, $a<\xi<b$，使得

$$\sum_{k=0}^{n-1} f^{(2N+2)}(a+kh+\theta_k h)=nf^{(2N+2)}(\xi).$$

于是，

$$E_{N+1}=-\frac{h^{2N+3}}{(2N+2)!}\int_0^1 [B_{2N+2}(t)-B_{2N+2}]$$

$$\times\sum_{k=0}^{n-1} f^{(2N+2)}(a+kh+th)dt$$

$$=-\frac{h^{2N+3}}{(2N+2)!}(-B_{2N+2}\cdot nf^{(2N+3)}(\xi))$$

$$=\frac{B_{2N+2}}{(2N+2)!}h^{2N+2}(b-a)f^{(2N+2)}(\xi).$$

这就证明了(5.6). Euler-Maclaurin 求和公式证毕.

将(5.16)代入上式，还可得到余项E_{N+1}的另一种表示式：

$$E_{N+1}=(-1)^{N+1}2\left(\frac{h}{2\pi}\right)^{2N+2}$$

$$\times(b-a)\zeta(2N+2)f^{(2N+2)}(\xi). \qquad (5.19)$$

§2 一些简单求积公式的导出

Euler-Maclaurin 求和公式(5.5)表明，在被积函数 $f(x)$ 充分可导的情况下，梯形和 $T(h)$ 具有(1.9)形式的渐近展开式，只含 h 的偶次幂项．因而可用 Romberg 外推法，来推算积分 $If=T(0)$ 的近似值 $T_m^{(i)}$．据(1.2)、(1.3)，其计算公式为

$$\begin{cases} T_0^{(i)}=T(h_i), \\ T_m^{(i)}=\dfrac{h_i^2T_{m-1}^{(i+1)}-h_{i+m}^2T_{m-1}^{(i)}}{h_i^2-h_{i+m}^2}. \end{cases} \tag{5.20}$$

$h_0, h_1, \cdots, h_m$ 不同，所得 $T_m^{(0)}$ 亦不同，我们记 $T(h_0, h_1, \cdots, h_m)=T_m^{(0)}$．许多求积公式，实质上就是

$$If\approx T(h_0, h_1, \cdots, h_m),$$

只是其中 $h_0, h_1, \cdots, h_m$ 的取法不同．换句话说，许多求积公式，实质上都是从梯形求积公式出发，应用 Romberg 外推法的结果．举例如下：

1° Simpson 求积公式．

设 n 是偶数，$h=(b-a)/n$，$h_0=2h$，$h_1=h$，$f_k=f(a+kh)$，则有

$$\begin{aligned} If\approx T_1^{(0)}&=\frac{(2h)^2T_0^{(1)}-h^2T_0^{(0)}}{(2h)^2-h^2}=\frac{1}{3}\{4T(h)-T(2h)\} \\ &=\frac{1}{3}\left\{4h\left(\frac{1}{2}f_0+f_1+f_2+\cdots+f_{n-2}+f_{n-1}+\frac{1}{2}f_n\right)\right. \\ &\qquad\left.-2h\left(\frac{1}{2}f_0+f_2+f_4+\cdots+f_{n-2}+\frac{1}{2}f_n\right)\right\} \\ &=\frac{h}{3}(f_0+4f_1+2f_2+\cdots+2f_{n-2}+4f_{n-1}+f_{2n}). \end{aligned}$$

这正是 Simpson 抛物线求积公式. 记右边的和式为 $S(h)$, 则有

$$If \approx S(h) = T(2h, h).$$

说明 Simpson 求积公式相当于由梯形求积公式作一步外推.

根据 Euler-Maclaurin 求和公式(5.5), 可立即得出 Simpson 公式的渐近展开式. 将(5.5)简记为

$$T(h) = If + \sum_{s=1}^{N} \tau_s h^{2s} + \tau_{N+1}(h) h^{2N+2}, \qquad (5.21)$$

其中

$$\tau_s = \frac{1}{(2s)!} B_{2s} [f^{(2s-1)}(b) - f^{(2s-1)}(a)],$$

$$\tau_{N+1}(h) = \frac{1}{(2N+2)!} (b-a) B_{2N+2} f^{(2N+2)}(\xi),$$

则有

$$\begin{aligned} S(h) - If &= \frac{1}{3} \{4[T(h) - If] - [T(2h) - If]\} \\ &= \frac{1}{3} \left\{ 4 \sum_{s=1}^{N} \tau_s h^{2s} + 4\tau_{N+1}(h) h^{2N+2} \right. \\ &\quad \left. - \sum_{s=1}^{N} \tau_s (2h)^{2s} - \tau_{N+1}(2h)(2h)^{2N+2} \right\} \\ &= \sum_{s=2}^{N} \frac{1}{3} (4 - 4^s) \tau_s h^{2s} + \tilde{\tau}_{N+1}(h) h^{2N+2}, \end{aligned}$$

其中

$$\begin{aligned} \tilde{\tau}_{N+1}(h) &= \frac{4}{3} \cdot \frac{b-a}{(2N+2)!} B_{2N+2} \\ &\quad \times [f^{(2N+2)}(\xi_1) - 4^N f^{(2N+2)}(\xi_2)]. \end{aligned}$$

类似于 Simpson 公式的推导, 可得下面的求积公式 2°、3°、4°.

2° Newton 八分之三求积公式.

设 n 是 3 的倍数，则有

$$If\approx N(h)=T(3h,\ h)=\frac{1}{8}\{9T(h)-T(3h)\}$$

$$=\frac{3}{8}h\{f_0+3f_1+3f_2+2f_3+3f_4+\cdots+3f_{n-1}+f_n\},$$

$$N(h)-If=\sum_{s=2}^{N}\frac{1}{8}(9-9^s)\tau_s h^{2s}+\tilde{\tau}_{N+1}h^{2N+2}.$$

当然，这里 $\tilde{\tau}_{N+1}$ 跟 Simpson 求积公式的 $\tilde{\tau}_{N+1}$ 不同.

3° Boole 求积公式. 设 n 是 4 的倍数，则有

$$If\approx B(h)=T(4h,\ 2h,\ h)$$

$$=\frac{1}{45}\{64T(h)-20T(2h)+T(4h)\}$$

$$=\frac{2}{45}h\{7f_0+32f_1+12f_2+32f_3+14f_4+\cdots+7f_n\},$$

$$B(h)-If=\sum_{s=3}^{N}\frac{\tau_s}{45}[64-20\times 4^s+16^s]h^{2s}+\tilde{\tau}_{N+1}h^{2N+2}.$$

4° Weddle 求积公式. 设 n 是 3 的倍数，则有

$$If\approx W(h)=T(3h,\ 2h,\ h)$$

$$=\frac{1}{10}\{15T(h)-6T(2h)+T(3h)\}$$

$$=\frac{3}{10}h\{f_0+5f_1+f_2+6f_3+f_4+5f_5+2f_6+\cdots+f_n\},$$

$$W(h)-If=\sum_{s=3}^{N}\frac{1}{10}(15-6\times 4^s+9^s)\tau_s h^{2s}$$

$$+\tilde{\tau}_{N+1}(h)h^{2N+2}.$$

上述求积公式都是梯形求积公式的线性组合. 由梯形求积公式的线性组合还可以得到其它求积公式，如

5° 中矩形求积公式.

$$If \approx U(h) = 2T(h/2) - T(h)$$

$$= h\{f_{\frac{1}{2}} + f_{\frac{3}{2}} + f_{\frac{5}{2}} + \cdots + f_{n-\frac{1}{2}}\}, \tag{5.22}$$

$$U(h) - If = \sum_{j=1}^{N}(2^{1-2j}-1)\tau_j h^{2j} + \tilde{\tau}_{N+1}(h)h^{2N+2}$$

$$= \sum_{j=1}^{N}\frac{(2^{1-2j}-1)}{(2j)!}B_{2j}h^{2j}If^{(2j)} + \tilde{\tau}_{N+1}(h)h^{2N+2}. \tag{5.23}$$

上述所有求积公式都是一百年以前提出来的. 我们在这里提及这些公式, 目的是说明, 它们都可以由梯形和 $T(h)$ 出发, 利用 Romberg 外推法或适当的线性组合得到, 而且它们误差的渐近展开式也可由 Euler–Maclaurin 求和公式推导出来.

其实, 不仅这些求积公式的误差渐近展开式可由 Euler–Maclaurin 求和公式推导出来, 复化的 Gauss–Legendre 求积公式也是如此. 事实上, 设 $f(x)$ 的 $2N+2$ 阶导数在区间 $[a, b]$ 上连续, 若在 $[a, b]$ 的 n 个相等小区间上应用 Gauss–Legendre 求积公式, 令 $h=(b-a)/n$, 那么

$$If = \sum_{l=1}^{n}\int_{a+(l-1)h}^{a+lh} f(x)dx$$

$$= \frac{h}{2}\sum_{l=1}^{n}\int_{-1}^{1} f\left(a+lh-\frac{h}{2}+\frac{h}{2}t\right)dt$$

$$\approx G(h) = \frac{h}{2}\sum_{l=1}^{n}\sum_{k=0}^{p} W_k f\left(a+lh-\frac{h}{2}+\frac{h}{2}\alpha_k\right).$$

这里 α_k 和 W_k 分别是 Gauss–Legendre 求积公式的节点和权系数. 将其中的 $f\left(a+lh-\frac{h}{2}+\frac{h}{2}\alpha_k\right)$ 按 Taylor 公式展开, 则得

$$G(h)=\frac{h}{2}\sum_{l=1}^{n}\sum_{k=0}^{p}w_k\left\{\sum_{j=0}^{2N+1}\frac{1}{j!}\left(\frac{h}{2}\alpha_k\right)^j f^{(j)}\left(a+lh-\frac{h}{2}\right)\right.$$
$$\left.+O(h^{2N+2})\right\}$$
$$=\sum_{l=1}^{n}\left\{\sum_{j=0}^{2N+1}\frac{h^{j+1}}{2^{j+1}j!}f^{(j)}\left(a+lh-\frac{h}{2}\right)\right.$$
$$\left.\times\sum_{k=0}^{p}w_k\alpha_k^j+O(h^{2N+3})\right\}.$$

注意 α_k 在$[-1, 1]$上的分布关于原点对称，而且对称节点处的权系数相同，可见 $j=2s+1$ 时

$$\sum_{k=0}^{p}w_k\alpha_k^{2s+1}=0.$$

又 $j=2s$ 时记

$$\beta_s=\frac{1}{2^{2s+1}(2s)!}\sum_{k=0}^{p}w_k\alpha_k^{2s},$$

则知

$$G(h)=\sum_{l=1}^{n}\left\{\sum_{s=0}^{N}\beta_s h^{2s+1}f^{(2s)}\left(a+lh-\frac{h}{2}\right)+O(h^{2N+3})\right\}$$
$$=\sum_{s=0}^{N}\beta_s h^{2s}\left\{h\sum_{l=1}^{n}f^{(2s)}\left(a+lh-\frac{h}{2}\right)\right\}+O(h^{2N+2}).$$

但由(5.23)知

$$h\sum_{l=1}^{n}f^{(2s)}\left(a+lh-\frac{h}{2}\right)=If^{(2s)}+\sum_{j=1}^{N}\frac{2^{1-2j}-1}{(2j)!}$$
$$\times B_{2j}h^{2j}If^{(2s+2j)}+O(h^{2N+2}).$$

代入 $G(h)$ 的表示式，可见

$$G(h)=\sum_{s=0}^{N}\bar{\tau}_s h^{2s}+O(h^{2N+2}),$$

其中 $\bar{\tau}_s$ 跟 h 无关. 可以证明，$\bar{\tau}_0=If$，$\bar{\tau}_1=\bar{\tau}_2=\cdots=\bar{\tau}_p=0$. 事实上，由 Gauss–Legendre 求积公式的余项公式

$$\sum_{k=0}^{p}w_k g(\alpha_k)=\int_{-1}^{1}g(t)dt=\frac{2^{2p+3}[(p+1)!]^4}{(2p+3)[(2p+2)!]^3}f^{(2p+2)}(\xi),$$

可知

$$G(h)-If=\frac{h}{2}\sum_{l=1}^{n}\left\{\sum_{k=0}^{p}w_k f\left(a+lh-\frac{h}{2}+\frac{h}{2}\alpha_k\right)\right.$$

$$\left.-\int_{-1}^{1}f\left(a+lh-\frac{h}{2}+\frac{h}{2}t\right)dt\right\}$$

$$=\frac{h}{2}\sum_{l=1}^{n}\frac{2^{2p+3}[(p+1)!]^4}{(2p+3)[(2p+1)!]^3}$$

$$\times f^{(2p+2)}(\xi_k)\left(\frac{h}{2}\right)^{2p+2}$$

$$=\frac{[(p+1)!]^4}{(2p+3)[(2p+2)!]^3}f^{(2p+2)}(\xi)h^{2p+2}.$$

比较 $G(h)$ 的最后两种表示式，便知 $\bar{\tau}_0=If$，$\bar{\tau}_1=\bar{\tau}_2=\cdots=\bar{\tau}_p=0$. 于是得到

$$G(h)=If+\sum_{s=p+1}^{N}\bar{\tau}_s h^{2s}+O(h^{2N+2}). \qquad (5.24)$$

§3 Romberg 积分法及其改进

上述求积公式 1°～4° 都是由梯形和 $T(h)$ 出发，按照 Romberg 外推法计算一、二步的结果，它们的误差阶数均比梯形求积公式的高. 由此自然想到，可将外推过程继续下去，直到 $|T_m^{(i)}-If|$ 小于容许误差限为止. 这种求积法称为 **Romberg 积分法**，是 1955年 Romberg 提出来的.

在外推过程中，参数(步长)数列 $\{h_i\}$ 通常采用第一种数列，即令 $h_i=h_0 2^{-i}$，而 $h_0=b-a$ 或 $b-a$ 的若干分之一. 此时，$(h_i/h_{i+m})^2=4^m$，Romberg 外推法的递推公式(1.3)变为

$$T_m^{(i)}=T_{m-1}^{(i+1)}+(T_{m-1}^{(i+1)}-T_{m-1}^{(i)})/(4^m-1); \qquad (5.25)$$

误差估计式(2.60)、(2.64)和(2.47)则分别变为

$$|T_{m+1}^{(i)}-If|\leqslant|T_m^{(i+1)}-T_m^{(i)}|/(1-4^{-m-1})$$
$$=|T_{m+1}^{(i)}-T_m^{(i)}|, \tag{5.26}$$
$$|T_m^{(i+1)}-If|\approx|T_m^{(i+1)}-T_m^{(i)}|/(4^m-1)$$
$$=|T_{m+1}^{(i)}-T_m^{(i+1)}|, \tag{5.27}$$
$$T_m^{(i)}-If=(-1)^m h_0^{2m+2}\{\tau_{m+1}+o(h_i^2)\}/$$
$$/2^{(m+2i)(m+1)}. \tag{5.28}$$

其中的(5.28)，根据 Euler-Maclaurin 求和公式(5.21)和 B_{2k} 的表示式(5.16)，还可改写为

$$T_m^{(i)}-If=\frac{(-1)^m B_{2m+2}h_0^{2m+2}}{2^{(m+2i)(m+1)}(2m+2)!}$$
$$\times\{f^{(2m+1)}(b)-f^{(2m+1)}(a)+o(h_i^2)\}$$
$$=\frac{(-1)^m B_{2m+2}h_0^{2m+2}(b-a)}{2^{(m+2i)(m+1)}(2m+2)!}$$
$$\times\{f^{(2m+2)}(\xi)+o(h_i^2)\}$$
$$=-\frac{\zeta(2m+2)h_0^{2m+2}(b-a)}{2^{m^2+(2i+3)m+2i+1}\pi^{2m+2}}$$
$$\times\{f^{(2m+2)}(\xi)+o(h_i^2)\}. \tag{5.29}$$

当然，在这些公式中，应有 $m\leqslant N$ 且 i 充分大. i 充分大的标志，是 $T_m^{(i)}$ 随 i 的增加而呈现单调性，或按(2.65)

$$D_m^{(i)}\equiv 4^{m+1}\frac{T_m^{(i+1)}-T_m^{(i)}}{T_m^{(i)}-T_m^{(i-1)}}\approx 1. \tag{5.30}$$

显然，上述公式都比一般 Romberg 外推法相应的公式简单而具体. 这是 Romberg 积分法通常采用第一种参数数列 H_1 的一个原因. 另一个原因，是在算出 $T(h_i)$ 之后，在需要缩小步长的时候，$T(h_{i+1})$ 比较好算. 事实上，在把区间[a, b]分成步长为 h_i 的小区间之后，再分成步长为 h_{i+1} 的小区间时，由于 $h_{i+1}=h_i/2$，新添节点就是原有小区间的中点. 所以，根据公式(5.22)，

$$T(h_{i+1})=\frac{1}{2}T(h_i)+\frac{1}{2}U(h_i)$$
$$=\frac{1}{2}T(h_i)+h_{i+1}\sum_{k=1}^{n/2}f(a+(2k-1)h_{i+1}),$$
(5.31)

其中 $n=(b-a)/h_{i+1}$. 这表明, $T(h_{i+1})$等于$T(h_i)$的一半加上新添节点处被积函数函数值之和与新步长 h_{i+1}的乘积. 因而计算$T(h_{i+1})$时, 只需计算新添节点处的函数值, 而不必重新计算$T(h_i)$所含的原有节点处函数值. 原有节点个数比新添节点个数还多1个, 这就使$T(h_{i+1})$包含的函数值计算工作量节省了一半以上.

不过, 这样每缩小一次步长, 新添节点个数成倍增加, 计算函数值的工作量也是成倍地增加. 对于比较复杂的被积函数, 这种工作量的成倍增加是十分可怕的. 为使函数值的计算工作量不致增加太快, 可采用第1章§4所说的第二、三种数列. 采用第三种数列H_3, 即令$h_i=h_0/(1+i)$, 每缩小一次步长, 所需函数值只增加一个, 可以说是增加速度最慢. 不过, 由于以前算过的函数值难以利用, 实际计算工作量得不到多大节省. 另外, 步长也不能缩得太小, 否则h_{i+1}/h_i逐渐接近于1, (2.73)中的M很大, 计算稳定性较差. 所以, 最好还是采用第二种参数数列H_2, 即

$$h_0,\ h_0/2,\ h_0/3,\ h_0/4,\ h_0/6,\ h_0/8,\ h_0/12,\ \cdots.$$

此时, 因为

$$T(h_{i+1})=\frac{1}{2}T(h_{i-1})+h_{i+1}\sum_{k=1}^{n/2}f(a+(2k-1)h_{i+1}),$$
(5.32)

其中$n=(b-a)/h_{i+1}$, 所以计算$T(h_{i+1})$时能充分利用$T(h_{i-1})$所用过的函数值, 新增的函数值计算工作量只是$T(h_{i-1})$的二

倍. 不过, 此时对应于(5.25)～(5.30)的公式要复杂一些, 因而计算机程序也要比第一种参数数列的复杂一点. 此外, 这时 $h_{i+1}/h_i \leqslant \frac{2}{3}$, 而对第一种参数数列 $h_{i+1}/h_i = \frac{1}{2}$,所以(2.73)中的 M 比较大, 计算稳定性稍差.

从(5.23)可见, 矩形和 $U(h)$ 也具有类似于梯形和 $T(h)$ 的渐近展开式, 而且 h^{2s} 的系数 $(2^{1-2s}-1)\tau_s$ 绝对值比 $T(h)$ 相应系数 τ_s 小. 所以, 我们也可由 $U(h)$ 出发来进行外推, 作外推表 $U_m^{(i)}$. 计算公式和程序跟 $T(h)$ 的差不多, $U_m^{(i)}$ 的误差比 $T_m^{(i)}$ 的还小一点. 注意(5.21)和(5.23)中 h^{2s} 的系数, 它们的符号完全相反. 这样由误差估计式(2.47)可见, 当 h_i 充分小时, $T_m^{(i)}$ 和 $U_m^{(i)}$ 相对 If 的误差具有相反的符号, 从而可作为 If 的渐近上下限(注意(5.22)与(2.61)相同, 这结论也可直接由第 2 章 § 5 的讨论得出). 于是, 我们也可同时计算 $T_m^{(i)}$ 和 $U_m^{(i)}$, 并利用(2.59)来估计误差. 此时 $T_m^{(i+1)}$ 还可根据(2.61)计算.

无论是梯形和 $T(h)$ 或中矩形和 $U(h)$, 它们的误差都只跟 h^2 同阶. 比起 § 2 导出的求积公式以及高斯型求积公式, 误差阶数都较低. 只有通过外推, 误差阶数才可能提高. 而为此得计算若干函数值. 因此, 为了节省计算函数值的工作量, 我们也可由高斯型求积公式出发进行外推. 注意(5.24), $G(h)$ 的渐近展开式中不含 h^2、h^4、…、h^{2p} 等项, 可把 $G(h_i)$ 看成 $G_p^{(i)}$, 继续计算 $m>p$ 时的 $G_m^{(i)}$.

对于一般的被积函数, 它在积分区间的不同部分上的变化快慢是不同的. 在某些部分, 用较大的步长、误差阶数较低的求积公式, 可以算出在这部分上积分的精确近似值. 而在另一些部分, 需用较小的步长, 较精确的求积公式, 才能得出

在这部分上积分的较精确近似值. 因此，在一个区间上利用同一步长、同一求积公式，是不划算的. 为了适应被积函数在不同部分变化极不相同的情况，Romberg 积分法可按如下步骤进行：

(a) 选定基本步长 h_0 和容许误差限 ε.

(b) 在区间 $[a, a+h_0]$ 上按 Romberg 积分法计算 $T_0^{(0)}$ 和 $T_1^{(0)}$，如果 $|T_1^{(0)}-T_0^{(0)}|<\varepsilon$，则接受 $T_1^{(0)}$(作为此区间上的积分值)，并将 h_0 放大1.5倍；否则计算 $T_2^{(0)}$，如果 $|T_2^{(0)}-T_1^{(0)}|<\varepsilon$，则接受 $T_2^{(0)}$；否则计算 $T_3^{(0)}$，如果 $|T_3^{(0)}-T_2^{(0)}|<\varepsilon$，则接受 $T_3^{(0)}$，并将 h_0 缩小为 $0.6h_0$；否则转向(d).

(c) 在接受 $T_1^{(0)}$ 或 $T_2^{(0)}$、$T_3^{(0)}$ 作为部分区间上的积分值之后，再在下一区间上继续按(b)进行，即将 a 换为上一区间的右端点(如果此时 $a+h_0>b$，则令 $h_0=b-a$)，然后转向(b). 这样直到整个积分区间积完为止.

(d)当(b)中 $T_3^{(0)}$ 不满足误差要求时，说明积分区间太大，可缩小 h_0 为$0.6h_0$，再转向(b). 如果 h_0 已缩小到很小的程度，$T_3^{(0)}$ 仍不满足误差要求，说明被积函数有问题，Romberg 积分法不能应用.

上述过程根据被积函数的特点，在积分区间不同部分上自动地改变求积公式的阶数($T_1^{(0)}$、$T_2^{(0)}$、$T_3^{(0)}$ 的误差阶数不同)和基本步长 h_0. 这种积分法称为**自适应积分法**.

我们在第 3 章 § 6 最后指出过，当 $T(h)$具有(2.12)那样的渐近展开式的时候，也可用有理式外推法，而且 $T_m^{(i)}$ 的误差往往比多项式外推法小. 由于梯形和 $T(h)$正好具有这种渐近展开式，即(5.21)，其中 $r=2$，所以我们也可放弃多项式外推法，改用有理式外推法，例如 Bulirsch–Stoer 有理式外推算法、Stoer 算法、Larkin 算法. 注意第 3 章 § 3 所列公式中的

h 应改为 h^2.

另外，当步长数列 $\{h_i\}$ 采用第一种数列 H_1 时，即令 $h_i = h_0 b^i$ 时，Euler-Maclaurin 求和公式(5.21)变为

$$T_i = T(h_i) = If + \sum_{s=1}^{N} \tau_s h_0^{2s} (b^{2s})^i + O(b^{2(N+1)i}).$$

这正是(4.1)形式的渐近展开式. 因此我们也可由梯形求积公式出发，利用 ε 算法推算积分值 If.

§4 反 常 积 分

应当注意，Euler-Maclaurin 求和公式(5.21)或(5.5)成立的条件，是积分区间有界，且被积函数 $f(x)$ 的 $2N+2$ 阶导数存在并且连续. 这样的积分称为正常积分. 如果积分是反常积分，即这两个条件不满足，前述外推法失去了基础，就未必成功了.

为了对反常积分应用外推法，可设法把反常积分的问题化为正常积分的问题. 例如在积分区间有限的情形，可采取下列办法:

1° 变量替换法. 利用变量替换把反常积分化为正常积分. 例如

$$\int_0^1 x^{p/q} g(x)dx \xlongequal{x=t^q} q\int_0^1 g(t^q) t^{p+q-1} dt,$$

$$\int_0^1 \frac{g(x)dx}{\sqrt{x(1-x)}} \xlongequal{x=\sin^2 t} 2\int_0^{\frac{\pi}{2}} g(\sin^2 t)dt,$$

$$\int_{-1}^1 \frac{g(x)dx}{\sqrt{1-x^2}} \xlongequal{x=\cos t} \int_0^{\pi} g(\cos t)dt. \qquad (5.33)$$

这里假定 $g(x)$ 的 $2N+2$ 阶导数连续，p、q 是正整数，从而上列三式右边都是正常积分.

2° 分项法．把被积函数 $f(x)$ 拆为两个函数之和：

$$f(x)=f_1(x)+f_2(x),$$

使 If_1 能直接算出，而 If_2 是正常积分． 例如，假定 $g(x)$ 的 $2N+3$ 阶导数连续，$0<p<1$，则

$$\int_0^1 x^{-p}g(x)dx$$

是反常积分．令

$$f_1(x)=x^{-p}\left[g(0)+\frac{x}{1!}g'(0)+\cdots\right.$$
$$\left.+\frac{x^{2N+2}}{(2N+2)!}g^{(2N+2)}(0)\right],$$
$$f_2(x)=x^{-p}g(x)-f_1(x),$$

则 $f_1(x)$ 很容易积分：

$$If_1=\frac{g(0)}{1-p}+\frac{g'(0)}{2-p}+\cdots+\frac{g^{(2N+2)}(0)}{(2N+2)!(2N+2-p)},$$

而 $f_2(x)$ 的 $2N+2$ 阶导数存在且连续，If_2 是正常积分．

3° 除去奇点法．即除去奇点(被积函数及其低阶导数不存在或不连续的点)的邻域，在剩下区间上积分．这时在剩下区间上的积分是正常积分． 采用这种办法时，应能估计在被除去部分上的积分，使其小于允许误差限．

有些反常积分，根本不考虑它的奇点，把它当成正常积分进行积分，也往往能获得成功． 例如在积分区间 $[-1,1]$ 两端是奇点的积分 $\int_{-1}^{1}f(x)dx$，如果在端点邻近的积分只占整个区间上积分的很小比重，可按正常积分进行外推． 为了避免端点处计算函数值的麻烦，可由中矩形求积公式出发进行外推． 为了进一步削弱端点附近积分的比重，还可先作余弦变换，令 $x=\cos t$，积分变为

$$\int_{-1}^{1} f(x)dx = \int_{0}^{\pi} f(\cos t)\sin t\, dt.$$

这时被积函数包含因式 $\sin t$，端点附近函数值对积分的影响就被缩小了．另外，也可除去奇点附近长为 h_i 的小区间，在剩下区间上进行积分，把积分值记为 $T(h_i)$，然后采用适当外推法推算整个区间上的积分值．

不过，不考虑反常积分的奇点，直接由矩形和或梯形和进行外推，这有些冒险．最好还是先研究奇点的出现对梯形和或矩形和渐近展开式的影响．为此，L. Fox 提出了一种研究方法．我们以下面只有一个奇点 $x=0$ 的积分为例，来说明他的方法．设有积分

$$\int_{0}^{1} f(x)dx = \int_{0}^{1} x^p g(x)dx, \tag{5.34}$$

其中 $f(x)=x^p g(x)$，$0<p<1$，$g(x)$ 充分连续可导．令

$$h=\frac{1}{n}, \quad F(x)=\int_{0}^{x} f(t)dt,$$

则按 Taylor 定理

$$f_0 = f_1 - hf_1' + \frac{h^2}{2!} f_1'' - \cdots. \tag{5.35}$$

$$\int_{0}^{h} f(x)dx = F_1 - F_0 = hF_1' - \frac{h^2}{2!} F_1'' + \frac{h^3}{3!} F_1''' - \cdots$$

$$= hf_1 - \frac{h^2}{2!} f_1' + \frac{h^3}{3!} f_1'' - \cdots.$$

由此二式消去 f_1'，即由第二式减去第一式的 $\frac{h}{2}$ 倍，得

$$\int_{0}^{h} f(x)dx = \frac{h}{2}(f_0+f_1) - \frac{h^3}{2\times 3!} f_1''$$

$$+ \frac{2h^4}{2\times 4!} f_1''' - \frac{3h^5}{2\times 5!} f_1^{(4)} + \cdots.$$

另一方面，按照 Euler-Maclaurin 求和公式(5.5)有

$$\int_h^1 f(x)dx=h\left\{\frac{1}{2}f_1+f_2+\cdots+f_{n-1}+\frac{1}{2}f_n\right\}$$

$$-\frac{1}{12}h^2(f'_n-f'_1)+\frac{1}{720}h^4(f'''_n-f'''_1)-\cdots.$$

将此二式相加，得

$$T(h)=\int_0^1 f(x)dx-\frac{h^2}{12}f'_1+\frac{h^3}{12}f''_1-\frac{29}{720}h^4f'''_1+\cdots$$

$$+\frac{h^2}{12}f'_n+\frac{h^4}{720}f'''_n-\cdots. \tag{5.36}$$

对于 $f(x)=x^pg(x)$, $0<p<1$,

$$f_1=h^pg(h)=g(0)h^p+\frac{g'(0)}{1!}h^{p+1}+\frac{g''(0)}{2!}h^{p+2}+\cdots,$$

$$h^2f'_1=h^2\left\{pg(0)h^{p-1}+(p+1)\frac{g'(0)}{1!}h^p\right.$$

$$\left.+(p+2)\frac{g''(0)}{2!}h^{p+1}+\cdots\right\}$$

$$=pg(0)h^{p+1}+(p+1)\frac{g'(0)}{1!}h^{p+2}$$

$$+(p+2)\frac{g''(0)}{2!}h^{p+3}+\cdots,$$

$$h^3f''_1=h^3\left\{p(p-1)g(0)h^{p-2}+(p+1)p\frac{g'(0)}{1!}h^{p-1}+\cdots\right\}$$

$$=p(p-1)g(0)h^{p+1}+(p+1)p\frac{g'(0)}{1!}h^{p+2}+\cdots,$$

……………………………………………………

代入(5.36)，则得

$$T(h)=If+\tau_1h^{p+1}+\tau_2h^{p+2}+\cdots+\bar{\tau}_1h^2+\bar{\tau}_2h^4+\cdots. \tag{5.37}$$

这里 τ_1, τ_2, $\cdots$, $\bar{\tau}_1$, $\bar{\tau}_2$, $\cdots$ 跟 h 无关，当然与(5.21)、(5.23)中系数不同. (5.37)说明，端点处 x^p 型奇点的出现，使梯形和

$T(h)$的渐近展开式中产生含 h^{p+1}、h^{p+2}、$\cdots$ 的误差项。

类似地可以证明，对于积分

$$\int_0^1 f(x)dx = \int_0^1 x^p(1-x)^q g(x)dx, \tag{5.38}$$

则有

$$T(h) = If + \tau_1 h^{p+1} + \tau_2 h^{p+2} + \cdots + \bar{\tau}_1 h^{q+1} + \bar{\tau}_2 h^{q+2} + \cdots \tag{5.39}$$

即两端同时出现 x^p、$(1-x)^q$ 型奇点，使梯形和 $T(h)$的渐近展开式中 h 的偶次幂项消失，而产生 h^{p+1}、h^{p+2}、$\cdots$、h^{q+1}、h^{q+2}、$\cdots$ 等等误差项．对于积分

$$\int_0^1 f(x)dx = \int_0^1 x^p \ln x g(x)dx, \tag{5.40}$$

有

$$T(h) = If + \tau_1 h^{p+1} + \tau_2 h^{p+1}\ln h + \tau_3 h^{p+2} + \tau_4 h^{p+2}\ln h + \cdots + \bar{\tau}_1 h^2 + \bar{\tau}_2 h^4 + \cdots. \tag{5.41}$$

对于积分

$$\int_0^1 f(x)dx = \int_0^1 x^p(1-x)^q \ln x \ln(1-x) g(x)dx, \tag{5.42}$$

有

$$\begin{aligned} T(h) = If &+ \tau_1 h^{p+1} + \tau_2 h^{p+1}\ln h + \tau_3 h^{p+2} \\ &+ \tau_4 h^{p+2}\ln h + \cdots + \bar{\tau}_1 h^{q+1} + \bar{\tau}_2 h^{q+1}\ln h \\ &+ \bar{\tau}_3 h^{q+2} + \bar{\tau}_4 h^{q+2}\ln h + \cdots. \end{aligned} \tag{5.43}$$

显然，在所有这些情况下，按照 Romberg 外推法、多项式外推法、Lagrange 外推法或有理式外推法进行外推，都不可能逐步消去低次误差项；是否能加速收敛，那是很侥幸的．由于矩形和 $U(h)$满足(5.21)，即 $U(h)=2T(h/2)-T(h)$，它也具有类似的渐近展开式，由矩形和 $U(h)$出发进行外推也有类

似的问题. 对照(2.16)、(2.27), 显然(5.37)、(5.39)的形式跟(2.16)相同, (5.41)、(5.43)的形式跟(2.27)相同. 因此, 对于相应的反常积分, 它们分别适宜于用 Bulirsh-Stoer 外推法与 Mühlbach 外推法进行外推. 在应用 Bulirsch-Stoer 外推法时, 如果被积函数比较复杂, p 和 q 的值可根据(2.65′)确定, 即当 $p<q$ 时

$$b^{p+1}\approx\frac{T(h_{i+1})-T(h_i)}{T(h_i)-T(h_{i-1})}, \tag{5.44}$$

$$b^{q+1}\approx\frac{T_1^{(i+1)}-T_1^{(i)}}{T_1^{(i)}-T_1^{(i-1)}}. \tag{5.45}$$

当然, 对于积分(5.37), 只用得着(5.44). 为了使 Bulirsch-Stoer 外推算法的进行稳妥可靠, 我们也可不断检查(2.65′)中的分式是否随 i 的增大而趋于定值.

在(2.16)型渐近展开式存在的情况下, 如果将步长数列 $\{h_i\}$ 取为第一种数列 H_1, 即

$$h_i=h_0b^i,$$

则(2.16)变为

$$\begin{aligned}T(h_i)=&\tau_0+\tau_1h_0^{r_1}(b^{r_1})^i+\tau_2h_0^{r_2}(b^{r_2})^i+\cdots\\&+\tau_Nh_0^{r_N}(b^{r_N})^i+O((b^{r_{N+1}})^i).\end{aligned}$$

这正是 (4.1) 型的渐近展开式. 因此, 在 Bulirsh-Stoer 外推法能用的情况下, 也能用 ε 算法. 这样, 对于诸如(5.34)、(5.38)形式的反常积分, 我们也能用 ε 算法推算积分 If. ε 算法用不着知道 r_1、r_2、$\cdots$ 等的具体数值, 所以比 Bulirsch-Stoer 外推法更通用.

以上讨论都是针对积分区间有限的反常积分来说的. 对于积分区间无限的反常积分, 也可类似地处理, 或者通过变量替换, 化为积分区间有限的积分.

§5 重 积 分

假定要计算二重积分

$$If=\iint_{\mathscr{D}} f(x,\ \tilde{x})dx\,d\tilde{x}. \tag{5.46}$$

其中 $\mathscr{D}$ 是矩形域：$a\leqslant x\leqslant b,\ \tilde{a}\leqslant\tilde{x}\leqslant\tilde{b}$. 把它化为二次积分，再用单变量求积公式(5.2)，得

$$If=\int_a^b dx\int_{\tilde{a}}^{\tilde{b}} f(x,\ \tilde{x})d\tilde{x}\approx\sum_{j=1}^{n} w_j\int_{\tilde{a}}^{\tilde{b}} f(x_j,\ \tilde{x})d\tilde{x}$$

$$\approx\sum_{j=1}^{n} w_j\sum_{k=1}^{\tilde{n}}\tilde{w}_k f(x_j,\ \tilde{x}_k).$$

令 $w_{jk}=w_j\tilde{w}_k$，则得

$$If\approx\sum_{j=1}^{n}\sum_{k=1}^{\tilde{n}} w_{jk}f(x_j,\ \tilde{x}_k). \tag{5.47}$$

这公式称为求积公式(5.2)的**乘积公式**，常用来求重积分的近似值．令 $h=(b-a)/n,\ \tilde{h}=(\tilde{b}-\tilde{a})/\tilde{n}$. 如果 $\tilde{h}/h=\mu$ 是常数，显然(5.47)右边跟 h 有关．由此自然使人想起，可用外推法推算积分值 If.

为了选择适当的外推算法，需研究乘积公式具有什么样的渐近展开式．这里只讨论梯形求积公式(5.3)导出的乘积公式

$$If\approx T(h)\equiv\mu h^2\sum_{j=0}^{n}{}''\sum_{k=0}^{\tilde{n}}{}'' f(x_j,\ \tilde{x}_k), \tag{5.48}$$

其中 Σ'' 表示首末两项只取一半的求和，

$$x_j=a+jh,\quad \tilde{x}_k=\tilde{a}+k\tilde{h}.$$

假定 $f(x,\ \tilde{x})$ 对 x、$\tilde{x}$ 的 $2N+2$ 阶偏导数连续，则据 Euler-Maclaurin 求和公式(5.5)有

$$
\begin{aligned}
T(h) &= h \sum_{j=0}^{n}{}'' \left\{ \int_{\tilde{a}}^{\tilde{b}} f(x_j, \tilde{x}) d\tilde{x} \right. \\
&\quad \left. + \sum_{s=1}^{N} \frac{\tilde{h}^{2s} B_{2s}}{(2s)!} \frac{\partial^{2s-1} f(x_j, \tilde{x})}{\partial \tilde{x}^{2s-1}} \Bigg|_{\tilde{x}=\tilde{a}}^{\tilde{x}=\tilde{b}} + o(\tilde{h}^{2N+2}) \right\} \\
&= \int_a^b dx \left\{ \int_{\tilde{a}}^{\tilde{b}} f(x, \tilde{x}) d\tilde{x} \right. \\
&\quad \left. + \sum_{s=1}^{N} \frac{\tilde{h}^{2s} B_{2s}}{(2s)!} \frac{\partial^{2s-1} f(x, \tilde{x})}{\partial \tilde{x}^{2s-1}} \Bigg|_{\tilde{x}=\tilde{a}}^{\tilde{x}=\tilde{b}} + O(\tilde{h}^{2N+2}) \right. \\
&\quad + \sum_{r=1}^{N} \frac{h^{2r} B_{2r}}{(2r)!} \frac{\partial^{2r-1}}{\partial x^{2r-1}} \left\{ \int_{\tilde{a}}^{\tilde{b}} f(x, \tilde{x}) d\tilde{x} \right. \\
&\quad \left. + \sum_{s=1}^{N} \frac{\tilde{h}^{2s} B_{2s}}{(2s)!} \frac{\partial^{2s-1} f(x, \tilde{x})}{\partial \tilde{x}^{2s-1}} \Bigg|_{\tilde{x}=\tilde{a}}^{\tilde{x}=\tilde{b}} \right\}_{x=a}^{x=b} + O(h^{2N+2}) \\
&= If + \sum_{j=1}^{N} \tau_j h^{2j} + O(h^{2N+2}). \qquad (5.49)
\end{aligned}
$$

其中的 τ_j 跟 h 无关，它的值跟(5.21)中的 τ_j 当然不同.

由(5.49)可见，(5.49)跟(5.21)形式上完全一样. 因此，由乘积公式(5.48)出发，可以像单变量的梯形求积公式一样，进行外推.

由(5.49)的推导可见，将二重积分改为多重积分，将(5.48)改为由其它单变量求积公式导出的乘积公式，如由中矩形求积公式、Gauss-Legendre 求积公式导出的乘积公式，将正常积分改为反常积分，那么跟(5.49)一样，乘积公式也具有类似于相应单变量求积公式的渐近展开式. 因此均可由乘积公式出发，采用类似于相应单变量求积公式的外推法，推算重积分的精确近似值.

注意(5.47)实质上是节点处被积函数函数值的线性组合. 由中矩形求积公式导出的乘积公式，其组合系数最简单，$w_{jk}=h\tilde{h}$，因此常用它作为外推的出发点. 高斯型求积公式能用较少的函数值算出较精确的结果，也可用作外推的出发点.

§6 数值微分

为了计算函数 $f(x)$ 的导数值，常常使用由插值多项式导出的数值微分公式，如

两点公式：

$$f'(0)\approx\frac{f(h)-f(0)}{h},\tag{5.50}$$

$$f'(0)\approx\frac{f(0)-f(-h)}{h};\tag{5.51}$$

三点公式：

$$f'(0)\approx\frac{1}{2h}[f(h)-f(-h)],\tag{5.52}$$

$$f'(0)\approx\frac{1}{2h}[-3f(0)+4f(h)-f(2h)];\tag{5.53}$$

二阶导数公式

$$f''(0)\approx\frac{1}{h^2}[f(h)-2f(0)+f(-h)].\tag{5.54}$$

从这些公式出发，可用外推法推算导数的精确近似值。

例如，对于(5.50)，令

$$G(h)=\frac{1}{h}[f(h)-f(0)],$$

则按 Taylor 公式有

$$G(h)=f'(0)+\frac{f''(0)}{2!}h+\frac{f'''(0)}{3!}h^2+\cdots$$
$$+\frac{f^{(N+1)}(0)}{(N+1)!}h^N+O(h^{N+1}).$$

这里 $G(h)$ 具有(2.12)形式的渐近展开式，$r=1$，因此可由数值微分公式(5.50)出发，利用多项式外推法、Lagrange 外推

算法、Bulirsch-Stoer 有理式外推法或者 ε 算法推算导数值 $f'(0)$.

又如，对于(5.52)，令

$$T(h)=\frac{1}{2h}[f(h)-f(-h)],$$

则根据 Taylor 公式有

$$T(h)=f'(0)+\frac{f'''(0)}{3!}h^2+\frac{f^{(5)}(0)}{5!}h^4+\cdots$$
$$+\frac{f^{(2N+1)}(0)}{(2N+1)!}h^{2N}+O(h^{2N+2}).$$

这里 $T(h)$ 具有(2.12)形式的渐近展开式，但 $r=2$，因此由(5.52)出发，也可利用 Romberg 外推法、Lagrange 外推法、Bulirsch-Stoer 有理式外推法或 ε 算法推算 $f'(0)$.

由外推法的误差分析知，$T_m^{(i)}$ 的误差阶数比 $G_m^{(i)}$ 的高. 因此，通常用中心差商公式(5.52)进行外推.

上述两种算法都是通过由插值多项式导出的数值微分公式进行外推. 其实，也可直接由插值多项式进行外推. 事实上，设 $P_m^{(i)}(x)$ 是函数 $f(x)$ 满足插值条件(2.1)的插值多项式,

$$P_m^{(i)}(x)=a_{m0}^{(i)}+a_{m1}^{(i)}x+a_{m2}^{(i)}x^2+\cdots+a_{mm}^{(i)}x^m, \quad (5.55)$$

则 $f(x)$ 在 $x=0$ 处的 s 阶导数值

$$f^{(s)}(0)\approx\frac{d^s}{dx^s}P_m^{(i)}(0)=s!a_{ms}^{(i)}. \quad (5.56)$$

另一方面，根据 Neville 线性插值公式(2.10)有

$$P_0^{(i)}(x)=f(x_i), \quad (5.57)$$

$$P_m^{(i)}(x)=\frac{(x-x_i)P_{m-1}^{(i+1)}(x)-(x-x_{i+m})P_{m-1}^{(i)}(x)}{x_{i+m}-x_i}. \quad (5.58)$$

对(5.58)两边逐次求导，得

$$\frac{d}{dx}P_m^{(i)}(x)=\frac{P_{m-1}^{(i+1)}(x)-P_{m-1}^{(i)}(x)+(x-x_i)\frac{d}{dx}P_{m-1}^{(i+1)}(x)}{x_{i+m}-x_i}$$

$$-\frac{(x-x_{i+m})\frac{d}{dx}P_{m-1}^{(i)}(x)}{x_{i+m}-x_i},$$

$$\frac{d^2}{dx^2}P_m^{(i)}(x)=\frac{2\frac{d}{dx}P_{m-1}^{(i+1)}(x)-2\frac{d}{dx}P_{m-1}^{(i)}(x)}{x_{i+m}-x_i}$$

$$+\frac{(x-x_i)\frac{d^2}{dx^2}P_{m-1}^{(i+1)}(x)-(x-x_{i+m})\frac{d^2}{dx^2}P_{m-1}^{(i)}(x)}{x_{i+m}-x_i},$$

....

$$\frac{d^s}{dx^s}P_m^{(i)}(x)=\frac{S\frac{d^{s-1}}{dx^{s-1}}P_{m-1}^{(i+1)}(x)-S\frac{d^{s-1}}{dx^{s-1}}P_{m-1}^{(i)}(x)}{x_{i+m}-x_i}$$

$$+\frac{(x-x_i)\frac{d^s}{dx^s}P_{m-1}^{(i+1)}(x)-(x-x_{i+m})\frac{d^s}{dx^s}P_{m-1}^{(i)}(x)}{x_{i+m}-x_i}.$$

令 $x=0$, 两边同除 $s!$, 则当 $0<s<m$ 时

$$a_{ms}^{(i)}=\frac{a_{m-1,s-1}^{(i+1)}-a_{m-1,s-1}^{(i)}+x_{i+m}a_{m-1,s}^{(i)}-x_ia_{m-1,s}^{(i+1)}}{x_{i+m}-x_i}. \tag{5.59}$$

当 $s=m$ 时, 注意 $\frac{d^m}{dx^m}P_{m-1}^{(i)}(x)=0$, 得

$$a_{mm}^{(i)}=\frac{a_{m-1,m-1}^{(i+1)}-a_{m-1,m-1}^{(i)}}{x_{i+m}-x_i},$$

而当 $s=0$ 时由(5.57)和(5.58)得

$$a_{00}^{(i)}=f(x_i), \tag{5.60}$$

$$a_{m0}^{(i)}=\frac{x_{i+m}a_{m-1,0}^{(i)}-x_ia_{m-1,0}^{(i+1)}}{x_{i+m}-x_i}. \tag{5.61}$$

显然，令 $a_{m,-1}^{(i)}=a_{m,m+1}^{(i)}=0$，则(5.59)当 $s=0, 1, \cdots, m$ 时均成立. 这样，利用公式(5.60)和递推公式(5.59)，便可求出 $a_{ms}^{(i)}$，从而按(5.56)计算 $f(x)$的导数 $f^{(s)}(0)$.

多项式外推法的目的，实质上是计算插值多项式 $P_m^{(i)}(x)$ 当 $x=0$ 时的值 $a_{m0}^{(i)}=P_m^{(i)}(0)$. 这里公式(5.60)、(5.59)不仅可用来计算 $a_{m0}^{(i)}$，也可用来计算 $P_m^{(i)}(x)$的其它系数 $a_{ms}^{(0)}$，从而推算 $f(x)$的导数值 $f^{(s)}(0)$，所以这种算法是多项式外推法的推广. 它是 1966 年 J. N. Lyness 和 B. Moler 提出来的，称为 **Lyness-Moler 外推算法.**

这种算法还可用来求解(2.3)型的线性代数方程组，也可用来计算函数 $f(x)$的导函数的积分. 事实上，根据 Euler-Maclaurin 求和公式(5.5)，

$$T(h)=If+\sum_{s=1}^{N}\frac{h^{2s}B_{2s}}{(2s)!}If^{(2s)}+O(h^{2N+2}), \quad (5.62)$$

其中 $$If^{(2s)}=\int_a^b f^{(2s)}(x)dx=f^{(2s-1)}(b)-f^{(2s-1)}(a).$$

在(5.62)中令 $x=h^2$，则右边对 x 的 s 阶导数当 $x=0$ 时的值为

$$\frac{s!}{(2s)!}B_{2s}If^{(2s)},$$

因此利用 Lyness-Moler 算法可求得这个数，从而求得导函数的积分 $If^{(2s)}$. 或者直接用 h^2 代替 x，令 $T_{ms}^{(i)}=a_{ms}^{(i)}$，(5.60)、(5.59)变为

$$\begin{cases} T_{m0}^{(i)}=T(h_i), & (5.63)\\ T_{ms}^{(i)}=\dfrac{T_{m-1,s-1}^{(i+1)}-T_{m-1,s-1}^{(i)}+h_{i+m}^2T_{m-1,s}^{(i)}-h_i^2T_{m-1,s}^{(i+1)}}{h_{i+m}^2-h_i^2}, & (5.64)\end{cases}$$

而 $f(x)$的导函数之积分

$$If^{(2s)}\approx(2s)!T_{ms}^{(i)}/B_{2s}.$$

$If^{(2s)}$ 在计算 $f(x)$的 Fourie 系数时需要用到.

第 6 章

外推法的应用(二)：函数方程数值解

外推法在微分方程、积分方程的数值解中已有广泛的应用．我们先从一个比较简单的例子——求解常微分方程初值问题的 Euler 折线法谈起．

§1　Euler 折线法与外推

设有常微分方程初值问题：

$$\begin{cases} y'=f(x, y), & x\in(a, b), \\ y(a)=y_0. \end{cases} \tag{6.1}$$

大家知道，它的解 $y(x)$ 一般不能用解析式表示；即使能表示，要计算实际所需的具体数值，也不一定方便．因此，常用的办法不是去求 $y(x)$ 的解析表达式，而是设法算出 $y(x)$ 在一系列点 $\{x_n\}$ $(x_n=a+nh,\ n=0,\ 1,\ 2,\ \cdots)$ 处的近似值 $\mathrm{y}_n\approx y(x_n)$，即所谓数值解．最简单的数值解法是 Euler 折线法，其计算公式为

$$\mathrm{y}_{n+1}=\mathrm{y}_n+hf(x_n,\ \mathrm{y}_n),\quad \mathrm{y}_0=y_0. \tag{6.2}$$

这种方法把求连续变量 $y(x)$ 的问题化为计算离散变量 $\{\mathrm{y}_n\}$ 的问题，因而是一种离散化方法．这里步长 h 是离散化参数．显然，对于固定的 x，按这种方法算出对应的 y_n(其中 $n=(x-x_0)/h$)跟 h 有关，可记为 $\mathrm{y}(x,\ h)$．由微分方程的理论可知，在 $f(x,\ y)$ 对 y 满足 Lipschitz 条件

$$|f(x,\ y_1)-f(x,\ y_2)|\leqslant L|y_1-y_2| \tag{6.3}$$

的情况下，所得解是收敛的，即

$$\lim_{h\to 0} \mathrm{y}(x,\ h) = y(x) \tag{6.4}$$

(后面将再作证明)．由此自然设想，根据若干 h 值算出的 $\mathrm{y}(x, h_0)$、$\mathrm{y}(x,\ h_1)$、$\mathrm{y}(x,\ h_2)$、…，利用外推法有可能算出 $y(x)$ 的更准确近似值．不过，究竟选择哪种算法最有效，还得看 $\mathrm{y}(x,\ h)$ 具有什么样的渐近展开式．

这种展开式必然同微分方程(6.1)的右部 $f(x,\ y)$ 和解 $y(x)$ 有关．这是因为，在 $f(x,\ y)=f(x)$ 的特殊情况下，方程(6.1)的解

$$\int_{x_0}^{x} f(t)dt$$

就是函数 $f(x)$ 的积分．由第5章§4和§1知道，在用步长 h 作数值积分时，所得近似式的渐近展开式的形式，同 $f(x)$ 有无奇点有关，而最大项数同 $f(x)$ 的可微性有关．特殊情况尚且如此，一般情况也就可想而知了．下面我们假定 $f(x,\ y)$ 没有奇点，充分可导．

按照(6.2)，$\mathrm{y}(x,\ h)=y_n$ 是根据 y_0 由 $y_1,\ y_2,\ \cdots,\ y_{n-1}$ 逐步计算出来的，所以要从(6.2)导出 $\mathrm{y}(x,\ h)$ 这种累积量、全程量的渐近展开式，就比较困难．但如果作一个局部化假定，即假设计算 y_n 时所需 y_{n-1} 没有误差、是真正解 $y(x)$ 对应的值 $y(x_{n-1})$，那么 y_n 的误差 $T_n=y(x_n)-y_n$ 的渐近展开式，利用 Taylor 公式就能很容易求出来．事实上，

$$\begin{aligned} T_{n+1} &= y(x_{n+1}) - y_{n+1} \\ &= y(x_{n+1}) - y(x_n) - hf(x_n,\ y(x_n)) \\ &= y(x+h) - y(x) - hy'(x) \\ &= \sum_{k=2}^{\infty} \frac{1}{k!} y^{(k)}(x) h^k. \end{aligned} \tag{6.5}$$

注意(6.5)，它也可以看成把初值问题(6.1)真正解代入公式(6.2)后两边之差，因此称为(6.2)的局部离散化误差.

局部离散化误差的渐近展开式，虽然严格说来不能作为应用外推法的基础，但由此可以推测，全程离散化误差的渐近展开式具有如下形式：

$$\varepsilon(x)=\mathrm{y}(x,\ h)-y(x)=\sum_{k=1}^{\infty}\tau_k(x)h^k,$$

从而

$$\mathrm{y}(x,\ h)=y(x)+\sum_{k=1}^{\infty}\tau_k(x)h^k. \tag{6.6}$$

这里 $\tau_k(x)$应怎样确定呢？试看 $\tau_k(x)$满足的条件. 将上式代入(6.2)，注意由 Taylor 公式有

$$z(x+h)-z(x)=\sum_{k=1}^{\infty}\frac{h^k}{k!}D^k z(x),$$

$$D=\frac{d}{dx}, \tag{6.7}$$

则得

$$\begin{aligned}
&\left(hD+\frac{h^2}{2!}D^2+\frac{h^3}{3!}D^3+\cdots\right)(y+\tau_1 h+\tau_2 h^2+\tau_3 h^3+\cdots)\\
&=hf(x,\ y+\tau_1 h+\tau_2 h^2+\tau_3 h^3+\cdots)\\
&=hf(x,\ y)+h\frac{\partial f}{\partial y}(\tau_1 h+\tau_2 h^2+\tau_3 h^3+\cdots)\\
&\quad+\frac{h}{2!}\frac{\partial^2 f}{\partial y^2}(\tau_1 h+\tau_2 h^2+\cdots)^2\\
&\quad+\frac{h}{3!}\frac{\partial^3 f}{\partial y^3}(\tau_1 h+\cdots)^3+\cdots,
\end{aligned} \tag{6.8}$$

其中

$$\frac{\partial^k f}{\partial y^k}=\frac{\partial^k f(x,\ y(x))}{\partial y^k}.$$

比较两边 h 的同次幂系数，得

$$\left.\begin{aligned}
&Dy=f(x,y),\\
&\left(D-\frac{\partial f}{\partial y}\right)\tau_1=-\frac{1}{2}D^2y,\\
&\left(D-\frac{\partial f}{\partial y}\right)\tau_2=-\frac{1}{2}D^2\tau_1-\frac{1}{3!}D^3y+\frac{1}{2!}\frac{\partial^2 f}{\partial y^2}\tau_1^2,\\
&\left(D-\frac{\partial f}{\partial y}\right)\tau_3=-\frac{1}{2}D^2\tau_2-\frac{1}{3!}D^3\tau_1-\frac{1}{4!}D^4y\\
&\qquad\qquad+\frac{\partial^2 f}{\partial y^2}\tau_1\tau_2+\frac{1}{3!}\frac{\partial^3 f}{\partial y^4}\tau_1^3,\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots
\end{aligned}\right\}\quad(6.9)$$

这里第一个方程是原来的微分方程(6.1)，其余则是逐步确定$\tau_1, \tau_2, \tau_3, \cdots$的线性微分方程. 注意

$$\mathrm{y}(x_0, h)=y_0=y(x_0),$$

便知

$$\tau_k(x_0)=0, \quad k=1, 2, \cdots. \qquad (6.10)$$

由此可知，$\tau_k(x)$作为线性微分方程初值问题的解，是完全确定的.

这样确定的$\tau_k(x)$，是否确使(6.6)成立？即是否真有

$$\mathrm{y}(x, h)-y(x)-\sum_{k=1}^{\infty}\tau_k(x)h^k=0?$$

为此，令

$$\begin{cases} S(x)=\sum\limits_{k=1}^{\infty}\tau_k(x)h^k,\\ \tilde{\varepsilon}(x)=\mathrm{y}(x, h)-y(x)-S(x)=\varepsilon(x)-S(x), \end{cases} \qquad (6.11)$$

则只需证明

$$\tilde{\varepsilon}(x)=0. \qquad (6.12)$$

为证(6.12)，需要用到如下的引理.

引理：设数列$\{\xi_n\}$满足不等式

$$|\xi_{n+1}|\leqslant A|\xi_n|+B, \tag{6.13}$$

其中 A 和 B 是非负常数，则 $n=1, 2, 3, \cdots$ 时有

$$|\xi_n|\leqslant A^n|\xi_0|+\begin{cases}\dfrac{A^n-1}{A-1}B, & A\neq 1,\\ nB, & A=1.\end{cases} \tag{6.14}$$

这可用数学归纳法证明. 事实上，$n=1$ 时(6.14)就是条件(6.13)当 $n=0$ 时的情形. 现设(6.14)当 $n=k$ 时成立，当 $n=k+1$ 时

$$|\xi_{k+1}|\leqslant A|\xi_k|+B\leqslant A\left(A^k|\xi_0|+\begin{cases}\dfrac{A^k-1}{A-1}B\\ kB\end{cases}\right)+B$$

$$=A^{k+1}|\xi_0|+\begin{cases}\dfrac{A^{k+1}-1}{A-1}B\\ (k+1)B.\end{cases}$$

这便证明了(6.14).

由此引理，立即可得如下两个有用的推论：

推论 1 如果数列 $\{\xi_n\}$ 满足等式

$$\xi_{n+1}=\xi_n+h\frac{\partial f(x_n, y(x_n))}{\partial y}\xi_n+\varphi, \tag{6.15}$$

其中 $$\left|\frac{\partial f}{\partial y}\right|\leqslant L, \quad |\varphi|\leqslant c_1h^p, \quad |\xi_0|\leqslant c_2h^q,$$

而 L, c_1, c_2, p, q 都是常数，则必存在常数 $\tilde{c}_1$ 和 $\tilde{c}_2$ 使

$$|\xi_n|\leqslant\tilde{c}_1h^{p-1}+\tilde{c}_2h^q. \tag{6.16}$$

事实上，由(6.15)，

$$|\xi_{n+1}|\leqslant(1+hL)|\xi_n|+c_1h^p.$$

按照引理，

$$|\xi_n|\leqslant(1+hL)^nc_2h^q+\begin{cases}\dfrac{(1+hL)^n-1}{hL}c_1h^p, & L\neq 0,\\ nc_1h^p, & L=0.\end{cases}$$

但 $nh=x_n-x_0\leqslant b-a$, $1+hL\leqslant e^{hL}$，故

$$|\xi_n| \leqslant c_2 e^{L(x_n-x_0)} h^q + c_1 h^{p-1} \begin{cases} \dfrac{e^{L(x_n-x_0)}-1}{L} \\ x_n - x_0 \end{cases}$$

$$\leqslant c_2 e^{L(b-a)} h^q + c_1 h^{p-1} \begin{cases} \dfrac{e^{L(b-a)}-1}{L} \\ b-a. \end{cases}$$

证毕.

推论 2 设初值问题(6.1)的解 $y(x)$ 的二阶导数连续，则存在常数 c，使

$$|\varepsilon(x_n)| = |\mathrm{y}(x_n, h) - y(x_n)| \leqslant ch. \tag{6.17}$$

证明 按照 Taylor 公式

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2} y''(\xi),$$

其中 ξ 在 x_n 与 x_{n+1}之间. 由(6.2)减去此式，得

$$\varepsilon(x_{n+1}) = \varepsilon(x_n) + h[f(x_n, \mathrm{y}(x_n, h)) - f(x_n, y(x_n))] - \frac{h^2}{2} y''(\xi).$$

注意条件(6.3)和 $y''(x)$ 的连续性，$|y''(x)| \leqslant M_2$，则

$$|\varepsilon(x_{n+1})| \leqslant |\varepsilon(x_n)| + hL|\varepsilon(x_n)| + \frac{h^2}{2} M_2.$$

由于 $\varepsilon(x_0) = 0$，故按推论 2

$$|\varepsilon(x_n)| \leqslant \frac{M_2}{2} h \begin{cases} \dfrac{e^{L(b-a)}-1}{L} \\ b-a. \end{cases}$$

证毕.

一般说来，如果近似解 $\mathrm{y}(x_n, h)$ 的误差

$$\varepsilon(x_n) = \mathrm{y}(x_n, h) - y(x_n)$$

满足关系 $$|\varepsilon(x_n)| \leqslant ch^p,$$

其中 c 和 p 都是常数，则称求 $\mathrm{y}(x_n, h)$ 的方法 p 阶收敛. 推

论2说明，Euler折线法1阶收敛。

利用引理和这两个推论，如果我们证明了

$$\tilde{\varepsilon}(x_{n+1})=\tilde{\varepsilon}(x_n)+h\frac{\partial f(x_n, y(x_n))}{\partial y}\tilde{\varepsilon}(x_n)+O(h^2)\tilde{\varepsilon}(x_n), \tag{6.18}$$

便可立即证明(6.12)。事实上，由(6.11)，

$$\tilde{\varepsilon}(x_n)=\varepsilon(x_n)-S(x_n),$$

而按(6.17)，$|\varepsilon(x_n)|=O(h)$，又显然 $S(x_n)=O(h)$，所以

$$\tilde{\varepsilon}(x_n)=O(h), \quad h^2\tilde{\varepsilon}(x_n)=O(h^3). \tag{6.19}$$

根据推论1并注意 $\tilde{\varepsilon}(x_0)=0$，$c_2=\tilde{c}_2=0$，便知

$$|\tilde{\varepsilon}(x_n)|=\tilde{c}_1O(h^{3-1})=O(h^2).$$

同理推知 $\qquad \tilde{\varepsilon}(x_n)=O(h^3)=O(h^4)=\cdots.$

这表明 $\qquad \tilde{\varepsilon}(x_n)=0.$

现在来证明(6.18)，或它的另一形式

$$\phi'\tilde{\varepsilon}\equiv\tilde{\varepsilon}(x_{n+1})-\tilde{\varepsilon}(x_n)-h\frac{\partial f}{\partial y}\tilde{\varepsilon}(x_n)=O(h^2\tilde{\varepsilon}(x_n)). \tag{6.18'}$$

事实上，记 $\tilde{\varepsilon}(x_n)=\tilde{\varepsilon}_n$，$\varepsilon(x_n)=\varepsilon_n$，$\mathrm{y}(x_n, h)=\mathrm{y}_n$，$y(x_n)=y_n$，则由(6.11)有

$$\begin{aligned}
\phi'\tilde{\varepsilon}&=\phi'(\mathrm{y}-y-S)=\phi'\mathrm{y}-\phi'(y+S)\\
&=\mathrm{y}_{n+1}-\mathrm{y}_n-h\frac{\partial f}{\partial y}\mathrm{y}_n\\
&\quad-\left[y_{n+1}+S_{n+1}-y_n-S_n-h\frac{\partial f}{\partial y}(y_n+S_n)\right]\\
&=\mathrm{y}_{n+1}-\mathrm{y}_n-hf(x_n, \mathrm{y}_n)\\
&\quad-[y_{n+1}+S_{n+1}-y_n-S_n-hf(x_n, y_n+S_n)]\\
&\quad+hf(x_n, \mathrm{y}_n)-hf(x_n, y_n+S_n)\\
&\quad-h\frac{\partial f}{\partial y}(\mathrm{y}_n-y_n-S_n).
\end{aligned}$$

注意 y_n 满足(6.2)，y_n+S_n 满足(6.8)，即满足把(6.6)代入(6.2)的结果

$$y_{n+1}+S_{n+1}=y_n+S_n+hf(x_n,\ y_n+S_n),$$

故得

$$\begin{aligned}\phi'\tilde{\varepsilon}&=hf(x_n,\ y_n)-hf(x_n,\ y_n+S_n)-h\frac{\partial f}{\partial y}(y_n-y_n-S_n)\\&=hf(x_n,\ y_n+S_n+\tilde{\varepsilon}_n)-hf(x_n,\ y_n+S_n)-h\frac{\partial f}{\partial y}\tilde{\varepsilon}_n\\&=h\tilde{\varepsilon}_n\left[\frac{\partial f(x_n,\ y_n+S_n+\theta\tilde{\varepsilon}_n)}{\partial y}-\frac{\partial f(x_n,\ y_n)}{\partial y}\right]\\&=h\tilde{\varepsilon}_n\frac{\partial^2 f(x_n,\ \xi)}{\partial y^2}(S_n+\theta\tilde{\varepsilon}_n),\end{aligned}$$

这里 $0<\theta<1$，ξ 在 y_n 和 $y_n+S_n+\theta\tilde{\varepsilon}_n$ 之间．注意到(6.19)及 $S_n=O(h)$，由上式立即可知

$$\phi'\tilde{\varepsilon}=O(h^2\tilde{\varepsilon}_n).$$

这便证明了(6.18)、(6.12)和(6.6)．

对于 Euler 折线法，既然 $y(x,\ h)$的渐近展开式具有(2.12)的形式，其中 $r=1$，$y(x,\ h)$相当于 $T(h)$，$y(x)$相当于 τ_0，我们便可利用多项式外推法、Lagrange 外推法、Bulirsh-Stoer 有理式外推法或 ε 算法，来推算 $y(x)$．

假定我们需要推算区间$[a,\ b]$上若干点 $\tilde{x}_j=a+jh_0$ 处 $y(x)$的值 $y(\tilde{x}_j)$，我们可以采用两种方式进行外推：一种，称为全程外推或消极外推，是用一种步长 h_k 计算出整个区间$[a,\ b]$上全部的 $y(\tilde{x}_j,\ h_k)=T_{0j}^{(k)}$ 之后，才在各点 $\tilde{x}_j$ 分别按外推法计算 $T_{mj}^{(i)}$．这里 T 的第二个下标 j 表示不同的点．这种外推方式占用存贮器较多，计算工作量也较大，但不改变基本离散化方法(本节是 Euler 折线法)的稳定性．另一种方式，称为局部外推或积极外推，是用步长 h_k 计算到 $y(\tilde{x}_1,\ h_k)$后，立即按外推法计算 $T_{m1}^{(i)}$ 以推算 $y(\tilde{x}_1)$，直到满足误差要求后，

才逐步地同样推算 $y(\tilde{x}_2)$, $y(\tilde{x}_3)$, $\cdots$. 这样每步均作外推的结果，可以看成把原始的离散化方法(这里是 Euler 折线法)变成另一种离散化方法,因而会改变稳定性.

下面我们以多项式外推算法的应用为例，说明局部外推会改变 Euler 折线法的稳定性.

我们知道,在数值分析理论中,用 h 为步长把某个单步法应用于试验方程

$$y'=\lambda y, \tag{6.20}$$

如果所得数值解 y_n 当 $n\to\infty$ 时趋于 0，则称这方法对于该 $\bar{h}=\lambda h$ 值绝对稳定；而使这方法绝对稳定的所有 $\bar{h}$ 值的全体，称为这方法的绝对稳定区域. 对 Euler 折线法(6.2),把它应用于(6.20),得

$$y_{n+1}=(1+\lambda h)y_n.$$

由此递推,可知

$$y_n=(1+\lambda h)^n y_0. \tag{6.21}$$

可见 $y_n\to 0$ 的充要条件是

$$|1+\lambda h|<1. \tag{6.22}$$

这就是 Euler 折线法(6.2)的绝对稳定区域. 当 $\bar{h}=\lambda h$ 为复数时,这是以 $\bar{h}=-1$ 为中心的单位圆内部;当 $\bar{h}$ 为实数时,这是区间$(-2, 0)$.

在将多项式外推法应用于 Euler 折线法时，假定 $h_k=h_0/2^k$, 各点 $\tilde{x}_j=a+jh_0$ 均计算m列，以 $T_{mj}^{(0)}$ 作为 $y(\tilde{x}_j)$的最后近似值,则由(6.21)

$$T_{0,j+1}^{(k)}=y(\tilde{x}_{j+1},\ h_k)=(1+\lambda h_0/2^k)^{2^k}T_{mj}^{(0)},$$

而由(2.24)

$$T_{m,j+1}^{(0)}=\sum_{k=0}^{m}c_{mk}^{(0)}(1+\lambda h_0/2^k)^{2^k}T_{mj}^{(0)}, \tag{6.23}$$

其中 $$c_{mk}^{(0)}=\prod_{j=0}^{m}{}'\frac{h_0/2^j}{h_0/2^j-h_0/2^k}=\prod_{j=0}^{m}{}'\frac{2^k}{2^k-2^j},$$

而当 $m=1, 2, 3, 4$ 时的具体数值可由(2.25)查得．例如

$$c_{10}^{(0)}=-1,\quad c_{11}^{(0)}=2,$$

$$c_{20}^{(0)}=\frac{1}{3},\quad c_{21}^{(0)}=-2,\quad c_{22}^{(0)}=\frac{8}{3}.$$

由(6.23)可见，绝对稳定性要求

$$\left|\sum_{k=0}^{m}c_{mk}^{(0)}(1+\lambda h_0/2^k)^{2^k}\right|<1. \tag{6.24}$$

这就是多项式外推法应用于 Euler 折线法后的绝对稳定区域．在 $m=1, 2$ 时，此即

$$\left|1+\lambda h_0+\frac{1}{2}\lambda^2h_0^2\right|<1,$$

$$\left|\frac{1}{3}(1+\lambda h_0)-2\left(1+\frac{1}{2}\lambda h_0\right)^2+\frac{8}{3}\left(1+\frac{1}{4}\lambda h_0\right)^4\right|<1.$$

在 $\bar{h}=\lambda h_0$ 为实数时，这是一些形如$(\alpha, 0)$的区间，其左端点 α 的数值如下表：

m	1	2	3	4	5	6
α	−2	−2.88	−4.24	−9.02	−10.90	−13.5

这说明，随着外推步数 m 的增加，绝对稳定区间越来越大，稳定性越来越好．

§2 离散化方法与外推

现在求解微分方程、积分方程的方法，大部分跟 Euler 折线法类似，不是求未知函数的解析表达式，而是求未知函数在若干点处的近似值，即所谓数值解．在未知函数为一元函数 $y(x)$ 的情形，这些点常常取为求解区间(a, b)上等距的点(称

为节点)：

$$x_i=a+ih,\ i=0,\ 1,\ \cdots,\ N,\ N=(b-a)/h;$$

在未知函数为二元函数 $u(x,\ y)$ 的情形，这些点常常取为求解区域上如下直线的交点 $(x_i,\ y_j)$（称为格点）：

$$x=x_i=a+ih,\ y=y_j=c+j\mu h,\ \mu\ 为常数.$$

未知函数为多元函数的情形类似. 未知函数的近似值 y_n $(\approx y(x_n))$ 或 $U_{ij}(\approx u(x_i,\ y_j))$ 满足的关系式因方法不同而不同. 它们往往是原始方程的某种近似，由此递推或解代数方程组，便能得出所需的 $\{\mathrm{y}_n\}$ 或 $\{U_{ij}\}$. 这类方法称为离散化方法.

下述方法都是离散化方法.

1° 求解常微分方程初值问题(6.1)的四种基本方法：

(1) Runge-Kutta 法

$$\begin{cases}\mathrm{y}_{n+1}=\mathrm{y}_n+\sum\limits_{r=1}^{R}w_rk_r,\\ k_r=hf\left(x_n+c_rh,\ \mathrm{y}_n+\sum\limits_{j=1}^{R}a_{rj}k_j\right)\quad(r=1,\ 2,\ \cdots,\ R)\end{cases}$$

其中 $w_r,\ c_r,\ a_{rj}$ 为常数. 若 $j\geqslant r$ 时有 $a_{rj}=0$，是显式法，否则是隐式法.

(2) 线性多步法

$$\mathrm{y}_{n+1}=\sum_{j=0}^{k-1}\alpha_j\mathrm{y}_{n-j}+h\sum_{j=-1}^{k-1}\beta_jf(x_{n-j},\ \mathrm{y}_{n-j}),$$

其中 α_j、β_j 为常数. $\beta_{-1}=0$ 时称显式法，否则称隐式法.

(3) 混合法

$$\mathrm{y}_{n+1}=\sum_{j=0}^{k-1}\alpha_j\mathrm{y}_{n-j}+h\sum_{j=0}^{k-1}\beta_jf(x_{n-j},\ \mathrm{y}_{n-j})+h\gamma f(x_{n+\mu},\ \mathrm{y}_{n+\mu}),$$

其中 $\alpha_j,\ \beta_j,\ \gamma,\ \mu$ 为常数，$0<\mu<1$.

(4) 单支法

$$y_{n+1}=\sum_{j=0}^{k-1}\alpha_j y_{n-j}+hf\left(\sum_{j=0}^{k-1}\beta_j y_{n-j}\right),$$

这里假定 $f(x, y)$ 与 x 无关，α_j、β_j 为常数.

上述第一种方法属于单步法. 单步法的一般形式为

$$y_{n+1}=y_n+h\phi(x_n, y_n, h). \tag{6.25}$$

其余三种方法都是多步法. 应用这些公式计算 y_{n+1}，需已知 y_n, y_{n-1}, …, y_{n-k+1} 等 k 个值，因而只有 $n\geqslant k-1$ 且已知 y_0, y_1, …, y_{k-1} 时才能开始计算. y_0, y_1, …, y_{k-1} 称为开始值或出发值. $y_0=y_0$是初值，由问题(6.1)本身给定；y_1, y_2, …, y_{k-1} 需用其它方法求出.

2° 有限差分法

这种方法是用导数的近似式替代微分方程中的导数，得出 y_n 或 U_{ij} 满足的关系式. 常用导数公式有

$$y'(x_n)\approx\frac{1}{2h}[y(x_{n+1})-y(x_{n-1})], \tag{6.26}$$

$$y'(x_n)\approx\frac{1}{h}[y(x_{n+1})-y(x_n)], \tag{6.27}$$

$$y''(x_n)\approx\frac{1}{h^2}[y(x_{n+1})-2y(x_n)+y(x_{n-1})]. \tag{6.28}$$

例如要解常微分方程边值问题

$$\begin{cases} y''=f(x, y, y'), & x\in(a, b) \\ y(a)=d_1, & y(b)=d_2. \end{cases} \tag{6.29}$$

设 $y_n\approx y(x_n)=y(a+nh)$，利用(6.26)和(6.28)得到 y_n 满足的代数方程组

$$\begin{cases} \dfrac{1}{h^2}(y_{n+1}-2y_n+y_{n-1})=f\left(x_n, y_n, \dfrac{1}{2h}(y_{n+1}-y_{n-1})\right) \\ \qquad (n=1, 2, \cdots, N-1) \\ y_0=d_1, \ y_N=d_2. \end{cases} \tag{6.30}$$

又如要解椭圆型偏微分方程的边值问题

$$\begin{cases}\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}=f(x,\ y,\ u), \quad a<x<b,\ c<y<d,\\ u|_{\Gamma}=\varphi(x,\ y), \quad \Gamma \text{ 为矩形域边界.}\end{cases} \tag{6.31}$$

利用(6.28)得到 $U_{ij}\approx u(x_i,\ y_j)$ 满足的代数方程组

$$\begin{cases}\dfrac{1}{h^2}(U_{i+1,j}-2U_{ij}+U_{i-1,j})+\dfrac{1}{\mu^2h^2}(U_{i,j+1}-2U_{ij}+U_{i,j-1})\\ \qquad =f(x_i,\ y_j,\ U_{ij}), \quad i=1,\ 2,\ \cdots,\ N-1,\\ U_{0j}=\varphi(a,\ y_j),\ U_{Nj}=\varphi(b,\ y_j), \quad j=1,\ 2,\ \cdots,\ M-1,\\ U_{i0}=\varphi(x_i,\ c),\ U_{iM}=\varphi(x_i,\ b).\end{cases} \tag{6.32}$$

其中 μ 为常数，$x_i=a+ih$, $y_j=c+j\tilde{h}$, $\tilde{h}/h=\mu$, M、N 为正整数，$h=(b-a)/N$, $\tilde{h}=(c-d)/M$.

3° 数值积分法

这种方法是用积分的近似式替代积分方程中的积分，得出 y_n 或 U_{ij} 满足的代数方程组．常用积分近似式是复化的梯形求积公式与 Simpson 公式：

$$\int_a^b f(x)dx\approx h\sum_{j=0}^{N}{}'' f(x_j), \tag{6.33}$$

$$\int_a^b f(x)dx\approx\frac{h}{3}[f(x_0)+4f(x_1)+2f(x_2)+\cdots +2f(x_{N-2})+4f(x_{N-1})+f(x_N)]. \tag{6.34}$$

这里 Σ'' 表示首末两项均取一半的求和，(6.34)中的 N 应为偶数．例如要解积分方程

$$y(x)-\int_a^b K(x,\ t,\ y(t))dt=0, \tag{6.35}$$

利用(6.33)得 $y_n\approx y(x_n)$ 满足的代数方程组

$$y_n-h\sum_{j=0}^{N}{}'' K(x_n, x_j, y_j)=0 \quad (n=0, 1, \cdots, N). \tag{6.36}$$

4° 有限单元法

这种方法首先把微分方程问题化为变分问题，然后用分段插值函数替代变分问题中的未知函数，根据极值原理得出 y_n 或 U_{ij} 满足的代数方程组．例如要求解常微分方程边值问题

$$\begin{cases}-(p(x)y')'+q(x)y=f(x), & a<x<b,\\ y(a)=y(b)=0.\end{cases} \tag{6.37}$$

由于这问题等价于在满足边界条件的函数中求函数$y(x)$,使得

$$\int_a^b [p(x)y'^2(x)+q(x)y^2(x)-2f(x)y(x)]^2dx$$

取极小值,代入

$$y(x)\approx\frac{x-x_i}{x_{i-1}-x_i}y_{i-1}+\frac{x-x_{i-1}}{x_i-x_{i-1}}y_i, \tag{6.38}$$

则变为求 $\{y_i\}$ 使下式取极小值的问题:

$$S=\sum_{i=1}^{N}\{k_{11}^{(i)}y_{i-1}^2+2k_{12}^{(i)}y_{i-1}y_i+k_{22}^{(i)}y_i^2-2b_1^{(i)}y_{i-1}-2b_2^{(i)}y_i\}, \tag{6.39}$$

其中$k_{11}^{(i)}$、$k_{12}^{(i)}$、$k_{22}^{(i)}$、$b_1^{(i)}$、$b_2^{(i)}$ 可根据 p, q, f 算出．由此按照极值原理，令 $\frac{\partial S}{\partial y_j}=0\,(j=1, 2, \cdots, N-1)$，则得 $\{y_i\}$ 满足的线性代数方程组．

上述各种方法得出的 y_n 或 U_{ij} 显然都同 h 有关，而且当 $h\to0$ 时应当趋于真正解 $y(x_n)$ 或 $u(x_i, y_j)$（设 x_n, x_i, y_j 的值固定）．由此自然想到，利用几个 h 算出的粗略近似值 y_n 或 U_{ij}，利用外推法进行外推，有可能得出更精确的近似值．

这种思想早在本世纪初便得到了应用. Richardson (1910)、(Runge(1912)、Gaunt(1927)等人把它应用于弦振动问题、矩形域上的 Laplace 方程问题、坝的压力问题、常微分方程的初值问题，并且获得了成功. 现在常常用两种步长的计算结果来估计误差、改进计算结果的方法，实质上是外推法，就是 1912 年 Runge 提出来的. 不过那时应用外推法，只是利用二、三个 h 算出的值进行外推，并没把这过程继续下去. 此外，那时在微分方程、积分方程以至数值积分中应用外推法的基础，是建立在局部离散化误差的渐近展开式之上的；全程离散化误差的渐近展开式，只是想当然的猜测，并没有严格的论证. 对此，Gragg 和 Stetter 1965 年进行了研究，并得出了相当一般的结果. 下节我们将加以介绍.

目前，对于一些具体的离散化方法，特别是一些新方法，例如上面所说的单支法和有限元法，其全程离散化误差的渐近展开式还缺乏研究. 因此，这些离散化方法是否可用外推法改进计算结果，还没有严格的理论基础. 但是，正如历史上的做法那样，我们也可以先把外推法应用起来，再逐步加强理论基础.

注意上述离散化方法，实际上可以不要节点等距或网格线等距之类的限制. 加上这类限制，是为了使近似解跟一个离散化参数 h 联系起来，便于应用外推法.

还需注意，应用外推法时可以采用局部或全程两种方式. 局部外推，会改变原始离散化方法的稳定性，因此适宜于能改进稳定性的离散化方法；全程外推不改变原始方法的稳定性，适宜于对稳定性要求比较苛刻的问题和稳定性较好的离散化方法. 例如对刚性常微分方程的初值问题，就适宜于对绝对稳定的离散化方法进行外推.

§3 全程误差渐近展开式

现在研究一般离散化方法的全程误差渐近展开式. 为了避免繁琐地逐一讨论各种离散化方法, 我们将使用 B 空间、算子等抽象概念. 下面首先介绍这些概念.

B 空间, 即 Banach 空间, 就是完全线性赋范空间. 线性空间, 是一种集合, 其中任一元素与数之积, 或者任意二元素之和, 仍然属于这集合. 如果这种集合的每一个元素 y 都有一个非负实数 $\|y\|$ (称为 y 的范数) 和它对应, 且满足下面三条件, 则称它为线性赋范空间:

(1) 当且仅当 $y=0$ 时 $\|y\|=0$;

(2) 对任意常数 α, $\|\alpha y\|=|\alpha|\,\|y\|$;

(3) $\|y+z\|\leqslant\|y\|+\|z\|$.

比如在区间 $[a, b]$ 上连续函数的全体 $C[a, b]$, 如规定其中函数 $y(x)$ 的范数为

$$\|y\|=\max_{a<x<b}|y(x)|, \tag{6.40}$$

就是线性赋范空间. 对于线性赋范空间, 可以规定两个元素 y 和 z 的"距离"为 $\|y-z\|$, 从而就能比较两个元素近似的程度, 讨论收敛性问题. 如果线性赋范空间的任意基本序列 $\{y_n\}$ (即对任意 $\varepsilon>0$ 存在整数 N, 使当 m, $n\geqslant N$ 时均有 $\|y_m-y_n\|<\varepsilon$ 的序列) 均有极限存在, 且属于这线性赋范空间, 则称这种线性赋范空间为完全线性赋范空间, 或 B 空间. 依 (6.40) 定义范数, $C[a, b]$ 就是 B 空间.

设有两个 B 空间 D_1 和 E_1, 如果存在一定规律, 使 D_1 中某一集合 D 的每一元素 y, 和 E_1 中某元素 z 相对应, 则称此规律为 D 上取值于 E_1 的算子, 记 $z=F(y)$. 对于 D 中任何

两元素 y_1、y_2 和任意常数 α_1、α_2，如有

$$F(\alpha_1 y_1+\alpha_2 y_2)=\alpha_1 F(y_1)+\alpha_2 F(y_2),$$

则称这种算子 F 为线性算子. 如果对 D 中某一元素 y，存在线性算子 A，使

$$\lim_{h\to 0}\left(\frac{1}{\|h\|}\|F(y+h)-F(y)-A(h)\|\right)=0,$$

则称 F 在 y 处 Fréchet 可微或 F 可微，并称线性算子 A 为 F 在 y 处的 F 导数，并记为 $F'(y)$. 在 F 为向量$(F_1, F_2, \cdots, F_m)^T$ 且 y 也为向量$(y_1, y_2, \cdots, y_n)^T$ 的时候，可以证明 $F'(y)$ 为 $m\times n$ 矩阵

$$F'(y)=A=(a_{ij}),\quad a_{ij}=\frac{\partial F_i(y)}{\partial y_j}.$$

利用上述概念，我们可以把一般微分方程、积分方程问题抽象成如下的算子方程组

$$\begin{cases} F(y)=0, & (6.41a)\\ R(y)=0. & (6.41b)\end{cases}$$

(6.41a)表示微分方程、积分方程，(6.41b)表示附加的初值条件、边值条件. F 和 R 两个算子分别定义于两个 B 空间 D_1 和 D_2，取值于 B 空间 E_1 和 E_2. 方程组(6.41)的解 y 就是 D_1 和 D_2 的公有部分 D 中，经算子 F 和 R 作用均变为 0 的元素. 我们假定它唯一. 同样地，把定义域剖分后，y 在网格点处的近似值 y 所满足的关系式(称为离散化算法)，也可抽象为算子方程组

$$\begin{cases} \phi_h(\mathrm{y})=0, & (6.42a)\\ P_h(\mathrm{y})=0, & (6.42b)\end{cases}$$

其中 ϕ_h、P_h 是同 h 有关的算子. 一般 y 是连续变量，y 是仅在格点处取值的离散变量. 由 y 在这些格点处取的值构成的离散变量，可看成为定义于 D 上的线性算子 Δ_h 的作用结果

$\Delta_h y$. ϕ_h、P_h 定义于 $\Delta_h D_1$、$\Delta_h D_2$. 方程组(6.42)的解 y 就是 $\Delta_h D$ 中，经 ϕ_h、P_h 作用均变为 0 的元素. 我们也假定它唯一.

算子方程组(6.42)自然应当是算子方程组(6.41)的某种近似. 故可以假定，$z\in D$ 时有

$$\begin{cases}\phi_h(\Delta_h z)=h^{n_1}\Delta_h\left\{F(z)+\sum_{\nu=p}^{N}h^\nu f_\nu(z)\right\}+O(h^{n_1+N+1}), & (6.43a)\\ P_h(\Delta_h z)=h^{n_2}\Delta_h\left\{R(z)+\sum_{\nu=p}^{N}h^\nu r_\nu(z)\right\}+O(h^{n_2+N+1}), & (6.43b)\end{cases}$$

这里 n_1, n_2 和 $p\geqslant 1$, $N\geqslant p$ 是适当的整数. 将(6.41)的解 y 代入，则有

$$\begin{cases}\phi_h(\Delta_h y)=h^{n_1}\Delta_h\sum_{\nu=p}^{N}h^\nu f_\nu(y)+O(h^{n_1+N+1}), & (6.44a)\\ P_h(\Delta_h y)=h^{n_2}\Delta_h\sum_{\nu=p}^{N}h^\nu r_\nu(y)+O(h^{n_2+N+1}). & (6.44b)\end{cases}$$

它称为算法(ϕ_h, P_h)的局部离散化误差.

在§1我们已经看到，由局部离散化误差的渐近展开式，可以推测全程误差

$$\varepsilon=\text{y}-\Delta_h y \tag{6.45}$$

的渐近展开式. 因此，在这里，由(6.44)可以设想

$$\varepsilon=\Delta_h\sum_{k=p}^{N}h^k\tau_k(y)+O(h^{N+1}),$$

即

$$\text{y}=\Delta_h\left\{y+\sum_{k=p}^{N}h^k\tau_k(y)+O(h^{N+1})\right\}. \tag{6.46}$$

τ_k 应怎样确定呢? 试看 τ_k 满足的条件，令

$$S=\sum_{k=p}^{N}h^k\tau_k(y),\quad \tilde{\varepsilon}=\varepsilon-S, \tag{6.47}$$

则显然有

$$S=O(h^p),\ \varepsilon=O(h^p),\ \tilde{\varepsilon}=O(h^p). \tag{6.48}$$

假定 F、f_ν、R、r_ν 在 y 附近的 M 阶 Frechet 导数存在，则

$$\phi_h^{(j)}(\Delta_h y)=h^{n_1}\Delta_h\left\{F^{(j)}(y)+\sum_{\nu=p}^{N}h^\nu f_\nu^{(j)}(y)\right\}+O(h^{n_1+N+1})$$
$$=O(h^{n_1}),\quad j=0,1,\cdots,M. \tag{6.49}$$

将(6.46)代入(6.42a)，则按 Taylor 公式，得

$$\begin{aligned}
0&=\phi_h(\Delta_h y+\Delta_h S+O(h^{N+1}))\\
&=\sum_{j=0}^{M-1}\frac{1}{j!}\phi_h^{(j)}(\Delta_h S+O(h^{N+1}))^j\\
&\quad+O(h^{n_1}(\Delta_h S+O(h^{N+1}))^M)\\
&=\sum_{j=0}^{M-1}\frac{1}{j!}\phi_h^{(j)}(\Delta_h S)^j+O(h^{n_1+N+1})+O(h^{n_1+Mp})\\
&=h^{n_1}\Delta_h\left\{F(y)+\sum_{\nu=p}^{N}h^\nu f_\nu(y)\right.\\
&\quad+\left(F'+\sum_{\nu=p}^{N}h^\nu f'_\nu\right)\left(\sum_{k=p}^{N}h^k\tau_k\right)\\
&\quad\left.+\sum_{j=2}^{M-1}\frac{1}{j!}\left(F^{(j)}+\sum_{\nu=p}^{N}h^\nu f_\nu^{(j)}\right)\left(\sum_{k=p}^{N}h^k\tau_k\right)^j\right\}\\
&\quad+O(h^{n_1+N+1})+O(h^{n_1+Mp}).
\end{aligned}\tag{6.50}$$

这里和下面我们将 Frechet 导数的求导处 y 略去不记．假定 $Mp\geqslant N+1$，$\nu=p,p+1,\cdots,2p-1$ 时 $g_\nu=0$，

$$\sum_{j=2}^{M-1}\frac{1}{j!}\left(F^{(j)}+\sum_{\nu=p}^{N}h^\nu f_\nu^{(j)}\right)\left(\sum_{k=p}^{N}h^k\tau_k\right)^j$$
$$=\sum_{\nu=2p}^{N}h^\nu g_\nu+O(h^{N+1}), \tag{6.51a}$$

(这里 g_ν 显然只同 y, τ_p, τ_{p+1}, $\cdots$, $\tau_{\nu-p}$ 有关)，注意 $F(y)=0$，则(6.50)变为

$$\begin{aligned}
0&=\phi_h(\Delta_h y+\Delta_h S+O(h^{N+1}))\\
&=h^{n_1}\Delta_h\sum_{\nu=p}^{N}\left(F'\tau_\nu+f_\nu+\sum_{\lambda=p}^{\nu-p}f'_\lambda\tau_{\nu-\lambda}+g_\nu\right)h^\nu\\
&\quad+O(h^{n_1+N+1}).
\end{aligned}\tag{6.52}$$

比较两边 h 的同次幂系数，得

$$F'\tau_k+f_k+\sum_{\lambda=p}^{k-p} f'_\lambda\tau_{k-\lambda}+g_k=0$$

$$(k=p,\ p+1,\ \cdots,\ N). \qquad (6.53a)$$

同理，将(6.46)代入(6.42b)，可得

$$R'\tau_k+r_k+\sum_{\lambda=p}^{k-p} r'_\lambda\tau_{k-\lambda}+t_k=0, \qquad (6.53b)$$

其中 t_k 由下式确定

$$\sum_{j=2}^{M-1}\frac{1}{j!}\left(R^{(j)}+\sum_{\nu=p}^{N}h^\nu r_\nu^{(j)}\right)\left(\sum_{k=p}^{N}\tau_k h^k\right)^j$$

$$=\sum_{\nu=2p}^{N}h^\nu t_\nu+O(h^{N+1}), \qquad (6.51b)$$

而 $\nu=p,\ p+1,\ \cdots,\ 2p-1$ 时 $t_\nu=0$. 这样，如果 y 有渐近展开式(6.46)，则 τ_k 满足方程组(6.53).

方程组(6.53)是 τ_k 的线性方程组，可写成

$$F'\tau_k=b_k,\quad R'\tau_k=c_k. \qquad (6.54)$$

在§1，相应方程组(6.9)和(6.10)，它们的解是唯一的. 所以，这里也可假定(6.53)或(6.54)的解唯一. 在此假定下，τ_k 是完全确定的.

现在我们来证明，反过来，如果 τ_k 满足(6.53)，则在一定条件下(6.46)成立. 就是说，我们有如下的定理：

Stetter 定理 1 假定

(a) (ϕ_h, P_h)具有局部离散化渐近展开式(6.43)；

(b) (6.43)中的算子 F、f_ν、R、r_ν 在 y 邻近 M 阶 Frechet 导数存在，且 $Mp\geqslant N+1$；

(c) (6.42)的解 y 是 p 阶收敛的，即存在常数 c，使

$$\|e\|\leqslant ch^p; \qquad (6.55)$$

(d) 方程组 $F'e=b$，$R'e=c$ 的解 e 唯一；

(e) (6.42)的解 y 是 n_1、n_2 稳定的，即方程组

$$\phi'_h e=\varphi,\quad P'_h e=\rho \tag{6.56}$$

的解 e 满足

$$\|e\|\leqslant S(h^{-n_1}\|\varphi\|_1+h^{-n_2}\|\rho\|_2), \tag{6.57}$$

其中 S 是常数，$\|\cdot\|$、$\|\cdot\|_1$、$\|\cdot\|_2$ 分别表示 $\Delta_h D$、$\Delta_h D_1$、$\Delta_h D_2$ 的范数.

那么，(6.42)的解 y 具有渐近展开式(6.46).

事实上，在(a)、(b)、(d)的假定，根据上面的讨论，方程组(6.53)及其解 $\tau_k(k=p,\ p+1,\ \cdots,\ N)$ 都是完全确定的. 按(6.47)定义 S 及 $\tilde{\varepsilon}$，则据假定(c)，(6.48)显然也成立. 这时为了证明(6.46)，我们只需证

$$\tilde{\varepsilon}=O(h^{N+1}). \tag{6.58}$$

为此，仿照§1，只需证明

$$\begin{cases}\phi'_h\tilde{\varepsilon}=h^{n_1}\{O(h^p\|\tilde{\varepsilon}\|)+O(h^{N+1})\}, & (6.59\text{a})\\ P'_h\tilde{\varepsilon}=h^{n_2}\{O(h^p\|\tilde{\varepsilon}\|)+O(h^{N+1})\}. & (6.59\text{b})\end{cases}$$

这是因为，由(6.48)，$\tilde{\varepsilon}=O(h^p)$，从而根据(6.59)和假定(e)，就可逐步推出 $\tilde{\varepsilon}=O(h^{2p})$，$\tilde{\varepsilon}=O(h^{3p})$，$\cdots$，$\tilde{\varepsilon}=O(h^{N+1})$.

现在我们来证明(6.59). 注意

$$\phi_h(\Delta_h y+\Delta_h S+\tilde{\varepsilon})=\phi_h(\mathrm{y})=0,$$

则由(6.49)、(6.48)、(6.44a)得

$$\begin{aligned}\phi'_h\tilde{\varepsilon}=&-[\phi_h(\Delta_h y+\Delta_h S+\tilde{\varepsilon})-\phi_h(\Delta_h y)-\phi'_h(\Delta_h S+\tilde{\varepsilon})]\\ &-\phi_h(\Delta_h y)-\phi'_h\Delta_h S\\ =&-\sum_{j=2}^{M-1}\frac{1}{j!}\phi_h^{(j)}(\Delta_h S+\tilde{\varepsilon})^j+O(h^{n_1+Mp})\\ &-\phi_h(\Delta_h y)-\phi'_h\Delta_h S.\end{aligned}$$

再注意(6.48)，则得

$$\phi_h'\tilde{\varepsilon} = -\sum_{j=0}^{M-1}\frac{1}{j!}\phi_h^{(j)}(\Delta_h S)^j + O(h^{n_1+p}\|\tilde{\varepsilon}\|) + O(h^{n_1+Mp})$$
$$= -\phi_h(\Delta_h y + \Delta_h S) + O(h^{n_1+Mp}) + O(h^{n_1+p}\|\tilde{\varepsilon}\|).$$

重复(6.52)的推导过程，并注意(6.53a)和条件(b)，便得

$$\phi_h'\tilde{\varepsilon} = -h^{n_1}\Delta_h\sum_{\nu=p}^{N}\left(F'\tau_\nu + f_\nu + \sum_{\lambda=p}^{\nu-p} f_\lambda'\tau_{\nu-\lambda} + g_\nu\right)h^\nu$$
$$+ O(h^{n_1+N+1}) + O(h^{n_1+Mp}) + O(h^{n_1+p}\|\tilde{\varepsilon}\|)$$
$$= O(h^{n_1+N+1}) + O(h^{n_1+p}\|\tilde{\varepsilon}\|).$$

这就证明了(6.59a)．同理可证(6.59b)．从而(6.58)、(6.46)得证．Stetter 定理 1 证毕．

应当注意，如果(6.43)中的 f_ν 和 r_ν 当 ν 为奇数时都是零算子，则(6.46)中的 τ_k 当 k 是奇数时也为 0．证明跟上面差不多．于是得到

Stetter 定理 2　设定理 1 的所有假定成立，但局部离散化误差渐近展开式(6.43)中的 f_ν 和 r_ν 当 ν 为奇数时是零算子，则全程离散化误差的渐近展开式(6.46)中只含 h 的偶次幂．

利用 Stetter 定理，可以推出许多离散化方法全程误差的渐近展开式．

例如，考虑一阶常微分方程组的初值问题:

$$\begin{cases} F(y) = y'(x) - G(y(x)) = 0, & a<x<b \\ R(y) = y(a) - y_0 = 0. \end{cases} \tag{6.60}$$

这里 y 是 l 维向量函数．

我们有 $D=D_1=C_l^{(1)}[a, b]$（区间$[a, b]$上的全体 l 维连续可微函数），$D_2=E_1=C_l[a, b]$（区间$[a, b]$上的全体 l 维连续函数），$E_2=R_l$（全体 l 维常量函数）．设$[a, b]_h=\{x_n: x_n=a+nh,\ n=0, 1, 2, \cdots, N,\ N=(b-a)/h\}$，则 Δ_h 把

$C_l[a, b]$的函数变为$[a, b]_h$上的函数.

如果采用 Euler 折线法作为离散化算法,则

$$\begin{cases}\phi_h(y)=(T_h-I)\mathrm{y}-hG(\mathrm{y})=0,\\ P_h(\mathrm{y})=\mathrm{y}(x_0)-y_0=0,\end{cases}\tag{6.61}$$

其中T_h是移位算子,$T_h\mathrm{y}(x_n)=\mathrm{y}(x_{n+1})$;$I$是恒等算子,$I\mathrm{y}=\mathrm{y}$. 假定$G$在(6.60)解$y$的邻域$M$阶连续可微,则$y\in C_l^{(M+1)}[a, b]$. 所以对$z\in C_l^{(M)}[a, b]$,按 Taylor 公式有

$$\begin{cases}\phi_h(z(x_n))=h\Big\{[z'(x)-G(z(x))]\\ \qquad\qquad+\displaystyle\sum_{\nu=1}^{N}\frac{h^\nu}{(\nu+1)!}\frac{d^{\nu+1}}{dx^{\nu+1}}z(x)\Big\}_{x=x_n}+O(h^{N+2}),\\ P_h(z)=h^0\{[z(x_0)-y_0]\}.\end{cases}$$

它对应于(6.43)的局部离散化渐近展开式,其中

$$p=1,\quad n_1=1,\quad n_2=0,$$

$$f_\nu=\frac{1}{(\nu+1)!}\frac{d^{\nu+1}}{dx^{\nu+1}},\quad r_\nu=0.$$

这说明 Stetter 定理 1 的条件(a)满足. 当$N\leqslant M-2$时,条件(b)显然也满足. 条件(c)和(c),由§1引理的推论2和(6.17)到向量的推广,也可见成立. 至于条件(d),要求如下方程组的解唯一,

$$e'-\frac{\partial G}{\partial y}e=b,\quad e(x_0)=c.$$

由于线性常微分方程组初值问题的解唯一,这要求自然满足. 于是,Stetter 定理 1 条件全部成立,§1所证全程误差渐近展开式(6.6)再次得到证明.

对于一般的单步法(6.25),要推出明显的展开式(6.43a)比较困难. 但实际上只需知道这样的展开式存在就够了,所以我们仍可仿照上面的推导,得出全程误差渐近展开式. 对

于多步法，令

$$\bar{\mathrm{y}}(x_n)=(\mathrm{y}(x_n),\ \mathrm{y}(x_{n+1}),\ \cdots,\ \mathrm{y}(x_{n+k-1}))^T,$$

则多步法形式上化为单步法，从而可按单步法应用 Stetter 定理。

§4 GBS 算 法

上节定理说明，一般离散化方法具有(6.46)形式的渐近展开式，在某些情况下，其中只含 h 的偶次幂．由于外推算法实质上是逐步消去全程误差中的低次项，所以从提高误差阶数的速度来看，外推法特别适宜于这种只含 h 偶次幂的情形．这情形的离散化方法称为对称方法．

怎样判断一个离散化方法是否对称方法呢？

对于求解常微分方程初值问题的单步法(6.25)，这很容易判断，只要看增量函数 ϕ 是否满足如下条件：

$$\phi(x+h,\ \mathrm{y}+h\phi(x,\ \mathrm{y},\ h),\ -h)=\phi(x,\ \mathrm{y},\ h). \tag{6.62}$$

事实上，对于(6.25)的解，上式表明

$$\phi(x_{n+1},\ \mathrm{y}_{n+1},\ -h)=\phi(x_n,\ \mathrm{y}_n,\ h). \tag{6.63}$$

因此，(6.25)等价于

$$\frac{1}{h}(\mathrm{y}_{n+1}-\mathrm{y}_n)=\frac{1}{2}[\phi(x_{n+1},\ \mathrm{y}_{n+1},\ -h)+\phi(x_n,\ \mathrm{y}_n,\ h)]. \tag{6.64}$$

令

$$\tilde{x}_n=x_n+\frac{h}{2}=x_{n+1}-\frac{h}{2},$$

则其局部离散化误差满足

$$l(\tilde{x}_n, h) = \frac{1}{h}[y(x_{n+1}) - y(x_n)]$$
$$- \frac{1}{2}[\phi(x_{n+1}, y(x_{n+1}), -h) + \phi(x_n, y(x_n), h)]$$
$$= \frac{1}{h}\left[y\left(\tilde{x}_n + \frac{h}{2}\right) - y\left(\tilde{x}_n - \frac{h}{2}\right)\right]$$
$$- \frac{1}{2}\left[\phi\left(\tilde{x}_n + \frac{h}{2}, y\left(\tilde{x}_n + \frac{h}{2}\right), -h\right)\right.$$
$$\left. + \phi\left(\tilde{x}_n - \frac{h}{2}, y\left(\tilde{x}_n - \frac{h}{2}\right), h\right)\right]$$
$$= l(\tilde{x}_n, -h). \tag{6.65}$$

故 $l(\tilde{x}_n, h)$ 只含 h 的偶次幂，从而按 Stetter 定理 2，(6.25)的全程误差渐近展开式只含 h 的偶次幂.

由此可以推出一个结论：如果(6.25)可改写为

$$\frac{1}{h}(y_{n+1} - y_n) = \psi(x_n, x_{n+1}, y_n, y_{n+1}, h), \tag{6.66}$$

则单步法(6.25)是对称方法. 因此显然，隐式中点法

$$y_{n+1} = y_n + hf\left(\frac{1}{2}(x_n + x_{n+1}), \frac{1}{2}(y_n + y_{n+1})\right) \tag{6.67}$$

及隐式梯形法

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \tag{6.68}$$

都是对称方法. 这两种方法都是隐式法，每算一步都必须解方程，比较费事，因而平常不用. 不过，这两种方法当 $\bar{h} = \lambda h$ 的实部为负数时是绝对稳定的，所以对于稳定性要求苛刻的刚性问题，也可用作离散化方法，然后进行全程外推.

对于求解常微分方程初值问题的线性多步法

$$y_{n+1} = \sum_{j=0}^{n-1} \alpha_j y_{n-j} + h \sum_{j=-1}^{n-1} \beta_j f(x_{n-j}, y_{n-j}), \tag{6.69}$$

对称性要求

$$\alpha_{k-j-2}=-\alpha_j,\quad \beta_{k-j-2}=\beta_j,\quad j=-1,\ 0,\ \cdots,\ k-1, \tag{6.70}$$

其中$\alpha_{-1}=0$. 事实上, 令

$$\tilde{x}_n=x_{n+1}-\frac{k}{2}h=x_{n-j}-\left(\frac{k}{2}-j-1\right)h,$$

则局部离散化误差(除以 h 以后)

$$\begin{aligned}
l(\tilde{x}_n,\ h)&=-\frac{1}{h}\left\{\sum_{j=-1}^{k-1}\alpha_j y(x_{n-j})\right.\\
&\quad\left.+h\sum_{j=-1}^{k-1}\beta_j f(x_{n-j},y(x_{n-j}))\right\}\\
&=-\frac{1}{h}\left\{\sum_{j=-1}^{k-1}\alpha_j y\left(\tilde{x}_n+\left(\frac{k}{2}-j-1\right)h\right)\right.\\
&\quad+h\sum_{j=-1}^{k-1}\beta_j f\left(\tilde{x}_n+\left(\frac{k}{2}-j-1\right)h,\right.\\
&\quad\left.\left.y\left(\tilde{x}_n+\left(\frac{k}{2}-j-1\right)h\right)\right)\right\}\\
&=\frac{1}{h}\left\{\sum_{j=-1}^{k-1}\alpha_{k-j-2}y\left(\tilde{x}_n+\left(\frac{k}{2}-j-1\right)h\right)\right.\\
&\quad-h\sum_{j=-1}^{k-1}\beta_{k-j-1}f\left(\tilde{x}_n+\left(\frac{k}{2}-j-1\right)h,\right.\\
&\quad\left.\left.y\left(\tilde{x}_n+\left(\frac{k}{2}-j-1\right)h\right)\right)\right\}.
\end{aligned}$$

令 $i=k-j-2$, 则

$$\begin{aligned}
l(\tilde{x}_n,\ h)&=\frac{-1}{(-h)}\left\{\sum_{i=-1}^{k-1}\alpha_i y\left(\tilde{x}_n-\left(\frac{k}{2}-i-1\right)h\right)\right.\\
&\quad-h\sum_{i=-1}^{k-1}\beta_i f\left(\tilde{x}_n-\left(\frac{k}{2}-i-1\right)h,\right.\\
&\quad\left.\left.y\left(\tilde{x}_n-\left(\frac{k}{2}-i-1\right)h\right)\right)\right\}\\
&=l(\tilde{x}_n,\ -h).
\end{aligned}$$

故 $l(\tilde{x}_n,\ h)$只含 h 的偶次幂, 从而据 Stetter 定理 2, (6.69)

的全程误差渐近展开式只含 h 的偶次幂.

一个真正有用的线性多步法至少应对 $y=1$ 和 $y=x$ 准确成立. 这样的方法称为相容的. 将 $y_n\equiv1$ 和 $y_n=nh$ 代入(6.69),得相容条件

$$1=\sum_{j=0}^{k-1}\alpha_j,\quad 1=-\sum_{j=1}^{k-1}j\alpha_j+\sum_{j=-1}^{k-1}\beta_j. \qquad (6.71)$$

在 $k=1$ 时,对称性条件(6.70)变为

$$-1=-\alpha_0,\quad \beta_{-1}=\beta_0,$$

而相容性条件(6.71)变为

$$1=\alpha_0,\quad 1=\beta_{-1}+\beta_0.$$

由此 $\alpha_0=1$, $\beta_{-1}=\beta_0=\frac{1}{2}$, (6.69)变为

$$y_{n+1}=y_n+\frac{h}{2}[f(x_{n+1},\ y_{n+1})+f(x_n,\ y_n)].$$

这正是(6.68). 这说明,对称且相容的线性单步法,只有隐式梯形法(6.68).

$k=2$ 时,对称条件(6.70)与相容条件(6.71)变为

$$-1=-\alpha_1,\quad \beta_{-1}=\beta_1,$$

$$1=\alpha_0+\alpha_1,\quad 1=-\alpha_1+\beta_{-1}+\beta_0+\beta_1.$$

由此 $\alpha_0=0$, $\alpha_1=1$, $\beta_{-1}=\beta_1$, $\beta_0=2-2\beta_1$. 可见显式方法($\beta_{-1}=0$)只有中点法

$$y_{n+1}=y_{n-1}+2hf(x_n,\ y_n). \qquad (6.72)$$

中点法跟一般的线性多步法一样,需要给定开始值 y_1 才能开始计算. 但随便给定 y_1, 可能破坏整个离散化方法的对称性. 例如令 $y_1=y(x_1)$, 即(6.1)真正解 $y(x)$ 在 $x=x_1$ 处的值,就会破坏对称性. Gragg 指出,令

$$y_1=y_0+hf(x_0,\ y_0), \qquad (6.73)$$

就可保持对称性. 事实上,

$$\begin{cases} \mathrm{y}_0 = y_0, \\ \mathrm{y}_1 = y_0 + hf(x_0 + y_0), \\ \mathrm{y}_{n+1} = \mathrm{y}_{n-1} + 2hf(x_n, \mathrm{y}_n) \end{cases} \tag{6.74}$$

等价于

$$\begin{cases} \mathrm{y}(x_0, h) = y_0, & (6.75\mathrm{a}) \\ \mathrm{y}(x_0 - h, h) = y_0 - hf(x_0, y(x_0, h)), & (6.75\mathrm{b}) \\ \mathrm{y}(x_n + h, h) = \mathrm{y}(x_n - h, h) + 2hf(x_n, \mathrm{y}(x_n, h)). & (6.75\mathrm{c}) \end{cases}$$

因此

$$\begin{aligned} \mathrm{y}(x_0, -h) &= y_0 = \mathrm{y}(x_0, h), \\ \mathrm{y}(x_1, -h) &= \mathrm{y}(x_0 + h, -h) = y_0 + hf(x_0, \mathrm{y}(x_0, -h)) \\ &= y_0 + hf(x_0, \mathrm{y}(x_0, h)) = \mathrm{y}(x_1, h), \\ \mathrm{y}(x_2, -h) &= \mathrm{y}(x_0, -h) + 2hf(x_1, \mathrm{y}(x_1, -h)) \\ &= \mathrm{y}(x_0, h) + 2hf(x_1, \mathrm{y}(x_1, h)) = \mathrm{y}(x_2, h). \end{aligned}$$

同理可得 $\mathrm{y}(x_n, h) = \mathrm{y}(x_n, -h)$.

因此在 $\mathrm{y}(x_n, h)$的渐近展开式中只含 h 的偶次幂.

将(6.74)应用于试验方程(6.20),即 $y' = \lambda y$ 时,有

$$\mathrm{y}_1 = (1 + \lambda h)\mathrm{y}_0, \tag{6.76}$$

$$\mathrm{y}_{n+1} = \mathrm{y}_{n-1} + 2\lambda h \mathrm{y}_n. \tag{6.77}$$

(6.77)是二阶常系数线性齐次差分方程,其解

$$\mathrm{y}_n = c_1 r_1^n + c_2 r_1^n, \tag{6.78}$$

而 r_1, r_2 是特征方程

$$r^2 = 1 + 2\lambda h r$$

的根,即

$$r_1 = \lambda h + \sqrt{\lambda^2 h^2 + 1}, \quad r_2 = \lambda h - \sqrt{\lambda^2 h^2 + 1}.$$

由条件 $\mathrm{y}_0 = y_0$ 和(6.76)可知

$$c_1 = \frac{\sqrt{\lambda^2 h^2 + 1} + 1}{2\sqrt{\lambda^2 h^2 + 1}} y_0, \quad c_2 = \frac{\sqrt{\lambda^2 h^2 + 1} - 1}{2\sqrt{\lambda^2 h^2 + 1}} y_0.$$

在 $\lambda h<0$ 时, $|r_1|<1$, $|r_2|>1$, 可见(6.78)中 $y_n\to\infty$(当 $n\to\infty$ 时). 这说明, 算法(6.74)是不稳定的. 为了削弱这种不稳定性, Gragg 提出, 令

$$\bar{y}_n=\frac{1}{4}(y_{n+1}+2y_n+y_{n-1}). \tag{6.79}$$

此时可以证明

$$\bar{y}_n=\frac{c_1}{4}(r_1^2+2r_1+1)r_1^{n-1}+r_2^{n-1}O(\lambda^4h^4). \tag{6.80}$$

在 $|\lambda h|<1$ 时, (6.80)中的第二项显然比(6.78)中的第二项小得多, 这就削弱了不稳定成分的影响.

将(6.74)与(6.79)结合起来, 得到如下所谓修改中点法:

$$\begin{cases} h=H/N,\ N \text{ 为偶数}, \\ y_0=y_0,\ y_1=y_0+hf(x_0,\ y_0), \\ y_{n+1}=y_{n-1}+2hf(x_n,\ y_n),\ n=1,\ 2,\ \cdots,\ N, \\ \bar{y}_N=\frac{1}{4}(y_{N+1}+2y_N+y_{N-1}). \end{cases} \tag{6.81}$$

此时, 根据上面的讨论, $\bar{y}_N$ 具有 h 偶次幂的渐近展开式, 而且具有较好的稳定性. 于是可以利用多项式外推法或 Bulirsch-Stoer 有理式外推法进行外推.

利用 Bulirsch-Stoer 有理式外推法对 $\bar{y}_N$ 作外推, 即按修改中点法计算 $\bar{y}_N$, 然后按 Bulirsch-Stoer 有理式外推法进行外推, 这种求解常微分方程初值问题的方法, 就是著名的 **Gragg-Bulirsch-Stoer 算法**, 简称 **GBS 算法**. Hull 等人通过多种问题的试验证明, 对于非刚性问题, 右端函数计算工作量不太大的时候, GBS 算法是一种最好的数值解法.

第 7 章

外推法的应用(三)：其他

§1　加速序列的收敛

外推法广泛应用于加速序列的收敛. 上面讲到在数值微积分、一般函数方程数值解中的应用，实际上就是加速序列 $\{T(h_i)\}$、$\{y(x, h_i)\}$ 的收敛，用几个 $T(h_i)$、$y(x, h_i)$ 的值来推算极限值 $T(0)$、$y(x, 0)$. 当然，外推算法还可用来加速其它序列的收敛.

例如，设有数列 $\{a_n\}$，其通项

$$a_n=\frac{1}{1\cdot 2\cdot 3}+\frac{3}{2\cdot 3\cdot 4}+\frac{5}{3\cdot 4\cdot 5}+\cdots+\frac{2n+1}{(n+1)(n+2)(n+3)}, \tag{7.1}$$

前几项为

$$a_0=\frac{1}{6},\quad a_1=\frac{7}{24},\quad a_2=\frac{3}{8},\quad a_3=\frac{13}{30},\quad a_4=\frac{10}{21},$$

而极限为 $\frac{3}{4}=0.75$. 把 ε 算法应用于此数列，即按(4.9)和(4.12)，令 $\varepsilon_{-1}^{(i)}=0$，$\varepsilon_0^{(i)}=a_i$，

$$\varepsilon_{m+1}^{(i)}=\varepsilon_{m-1}^{(i+1)}+\left(\varepsilon_m^{(i+1)}-\varepsilon_m^{(i)}\right)^{-1}, \tag{7.2}$$

得到如下的表 7.1.

表中第三列 $\varepsilon_2^{(i)}$ 可看成用 Aitken Δ^2 加速法(4.10)计算的结果，其中 $\varepsilon_2^{(2)}=\frac{116}{195}\approx 0.5949$，它近似极限值 0.75 的精确

表 7.1

$\frac{1}{6}(\approx 0.1667)$				
	8			
$\frac{7}{24}(\approx 0.2916)$		$\frac{13}{24}(\approx 0.5417)$		
	12		48	
$\frac{3}{8}(\approx 0.3750)$		$\frac{41}{72}(\approx 0.5694)$		$\frac{11}{16}(=0.6875)$
	$\frac{120}{7}$		$\frac{6720}{119}$	
$\frac{13}{30}(\approx 0.4333)$		$\frac{116}{195}(\approx 0.5949)$		
	$\frac{70}{3}$			
$\frac{10}{21}(\approx 0.4762)$				

度，相当于 $a_{10}=\frac{31}{52}\approx 0.5962$. 注意它只是用 a_2、a_3、a_4 计算的结果，说明 Aitken Δ^2 加速法确实加速了数列(7.1)的收敛. 表中 $\varepsilon_4^{(0)}=\frac{11}{16}=0.6875$, 是用 a_0, a_1, a_2, a_3, a_4 算出的，相当于用 $a_{29}\approx 0.6880$ 近似极限值. 可见 ε 算法加速收敛的效果不错.

将多项式外推法应用于数列(7.1)，即按公式(2.13)、(2.14)，令 $T_0^{(i)}=a_i$,

$$T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\dfrac{i+m+1}{i+1}-1}. \tag{7.3}$$

这里已设 $r=1$, $h_i=1/(i+1)$. 我们得到表 7.2.

表 7.2 中的

$$T_2^{(2)}=\frac{1187}{1680}\approx 0.7065, \quad T_4^{(0)}=\frac{7407}{10080}\approx 0.73482,$$

分别是用 a_2、a_3、a_4 和 a_0、a_1、a_2、a_3、a_4 算出的，但近似极限值 0.75 的精度，分别相当于

$$a_{43}\approx 0.7068, \quad a_{128}\approx 0.73476.$$

表 7.2

$\frac{1}{6}$				
$\frac{7}{24}$	$\frac{5}{12}(\approx 0.4167)$			
$\frac{3}{8}$	$\frac{13}{24}(\approx 0.5417)$	$\frac{29}{48}(\approx 0.6042)$		
$\frac{13}{30}$	$\frac{73}{120}(\approx 0.6083)$	$\frac{27}{40}(\approx 0.6750)$	$\frac{503}{720}(\approx 0.6986)$	
$\frac{10}{21}$	$\frac{68}{105}(\approx 0.6476)$	$\frac{1187}{1680}(\approx 0.7065)$	$\frac{3667}{5040}(\approx 0.7276)$	$\frac{7407}{10080}(\approx 0.7348)$

可见多项式外推法加速数列(7.1)的收敛，比 ε 算法的效果更好.

将 ρ 算法应用于数列(7.1)，即按公式(3.36)，取 $h_i=i+1$，令 $T_0^{(i)}=a_i$, $T_{-1}^{(i)}=0$,

$$T_m^{(i)}=T_{m-2}^{(i+1)}+\frac{m}{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}. \qquad (7.4)$$

这实际上是带权的 ε 算法. 我们得到表 7.3. 其中 $T_4^{(0)}$ 等于

表 7.3

$\frac{1}{6}$				
	8			
$\frac{7}{24}$		$\frac{19}{24}(\approx 0.7917)$		
	12		-96	
$\frac{3}{8}$		$\frac{55}{72}(\approx 0.7639)$		$\frac{3}{4}(=0.75)$
	$\frac{120}{7}$		-384	
$\frac{13}{30}$		$\frac{59}{78}(\approx 0.7564)$		
	$\frac{70}{3}$			
$\frac{10}{21}$				

极限值0.75.

把Stoer外推算法应用于数列(7.1)，即按公式(3.33)、(3.34)，令 $T_{-1}^{(i)}=0$, $T_0^{(i)}=a_i$, $h_i=i+1$,

$$T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\frac{i+m+1}{i+1}\theta_m^{(i)}-1}, \tag{7.5}$$

$$\theta_m^{(i)}=\frac{T_{m-1}^{(i)}-T_{m-1}^{(i+1)}}{T_{m-1}^{(i+1)}-T_{m-2}^{(i+1)}},$$

我们得到表7.4. 其中 $T_3^{(0)}$ 已等于极限值0.75.

表 7.4

$\frac{1}{6}$				
	$\frac{7}{6}(\approx 1.1667)$			
$\frac{7}{24}$		$\frac{19}{24}(\approx 0.7917)$		
	$\frac{7}{8}(\approx 0.8750)$		$\frac{3}{4}$	
$\frac{3}{8}$		$\frac{55}{72}(\approx 0.7639)$		$\frac{3}{4}(=0.75)$
	$\frac{13}{16}(\approx 0.8125)$		$\frac{3}{4}$	
$\frac{13}{30}$		$\frac{59}{78}(\approx 0.7564)$		
	$\frac{26}{33}(\approx 0.7879)$			
$\frac{10}{21}$				

比较上述四种外推算法计算的结果，可见对数列(7.1)，ρ 算法与Stoer外推算法的效果较好，而Stoer外推法最好. 后两种算法都是有理式外推法. 它们之所以比 ε 算法与多项式外推法好，而且能算出极限的准确值，是因为数列(7.1)具有有理式展开式：

$$a_n=\frac{3}{4}-\frac{4n}{2(n+1)(n+3)}. \tag{7.6}$$

对于其它数列，加速收敛的效果，不一定是Stoer外推算法最好、ε 算法最差. 例如对于数列

$$x_n = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots + \frac{(-1)^{n-1}}{n}, \qquad (7.7)$$

ε 算法的效果最好. 结果见表 7.5. x_n 的极限值是

$$\ln 2 \approx 0.69314718.$$

得到的

$$\varepsilon_4^{(0)} = \frac{52}{75} \approx 0.693333,$$

相当于 x_{5369} 达到的精度. 而多项式外推法、Stoer 外推法、ρ 算法均不能加速收敛.

表 7.5

1				
	-2			
$\frac{1}{2}(=0.5)$		$\frac{7}{10}(=0.7)$		
	3		-102	
$\frac{5}{6}(\approx 0.8333)$		$\frac{29}{42}(\approx 0.6905)$		$\frac{52}{75}(\approx 0.693333)$
	-4		248	
$\frac{7}{12}(\approx 0.5833)$		$\frac{25}{36}(\approx 0.6944)$		
	5			
$\frac{47}{60}(\approx 0.7833)$				

一般说来, 象(7.7)这样的振荡数列, 都不适宜于用多项式外推法或有理式外推法. 这是因为, 适于应用这两类算法外推的数列 $\{T(h_i)\}$, 其渐近展开式具有(2.16)的形式, 即有

$$T(h) = \tau_0 + \tau_1 h^{r_1} + \tau_2 h^{r_2} + \cdots + \tau_N h^{r_N} + \tau_{N+1}(h) h^{r_{N+1}},$$

可见 $h \to 0$ 时 $T(h)$ 单调地趋于 $T(0)$. 适于应用 ε 算法的数列 $T_k = T(k)$, 具有(4.1)型的渐近展开式, 即

$$T_k = \tau_0 + \tau_1 \lambda_1^k + \tau_2 \lambda_2^k + \cdots + \tau_N \lambda_N^k + O(\lambda_N^k).$$

当 λ_1 是负数时, T_k 便是振荡的数列.

§2 级数求和与特殊函数的求值

数列(7.1)与(7.7)都可看成无穷级数的部分和. 因此上面的讨论说明, 外推算法可应用于级数的求和. 许多特殊函数可借用级数表示, 例如

$$\ln(1+x)=x-\frac{x^2}{2}+\frac{x^3}{3}-\frac{x^4}{4}+\cdots+(-1)^{n-1}\frac{x^n}{n}+\cdots \tag{7.8}$$

因而外推算法也可用来求这些特殊函数的值. 上面数列(7.7)就是函数(7.8)当 $x=1$ 时的值. 值得注意的是, 当 $x=2$ 时, (7.8)右边的级数发散, 但应用 ε 算法, 仍可用来求 ln 3 的值. 计算结果见表7.6. 注意 $\ln 3=1.0986123$, $s_4^{(0)}=\frac{76}{69}\approx 1.1014$ 已相当精确.

表 7.6

2				
	$-\frac{1}{2}$			
0		$\frac{8}{7}(\approx 1.1429)$		
	$\frac{3}{8}$		$-\frac{51}{4}$	
$\frac{8}{3}(\approx 2.6667)$		$\frac{16}{15}(\approx 1.0667)$		$\frac{76}{69}(\approx 1.1014)$
	$-\frac{1}{4}$		16	
$-\frac{4}{3}(\approx -1.3333)$		$\frac{44}{39}(\approx 1.1282)$		$\frac{260}{237}(\approx 1.0970)$
	$\frac{5}{32}$		$-\frac{515}{32}$	
$\frac{76}{15}(\approx 5.0667)$		$\frac{16}{15}(\approx 1.0667)$		$\frac{1996}{1815}(\approx 1.0997)$
	$-\frac{3}{32}$		$\frac{453}{32}$	
$-\frac{28}{5}(=-5.6)$		$\frac{108}{95}(\approx 1.1368)$		
	$\frac{7}{128}$			
$\frac{444}{35}(\approx 12.6857)$				

§3 方程求根

求解方程

$$f(x)=0 \tag{7.9}$$

的基本方法，是求数列 $\{x_n\}$，使 x_n 逐步趋近于(7.9)的根 α. 对于数列 $\{x_n\}$，自然可象 §1 那样，用外推法加速其收敛.

在线性收敛的情况下，即当

$$x_{n+1}-\alpha\approx c(x_n-\alpha),\quad c\neq 0 \tag{7.10}$$

时，可用 Aitken Δ^2 法加速收敛. 这是因为，(7.10)可改写为

$$c(x_n-\alpha)-(x_{n+1}-\alpha)\approx 0,$$

即近似满足 ε 算法性质 2° 的条件，其中 $m=1$，因而

$$\varepsilon_2^{(i)}\approx\alpha.$$

方程(7.9)的根 α，可看成 $y=f(x)$的反函数 $x=\varphi(y)$当 $y=0$ 时的值，即 $\alpha=\varphi(0)$. 因此，在单根 α 附近，当 $f(x)$充分可微时，根据隐函数的理论，有

$$x=\alpha+\sum_{k=1}^{N}\frac{1}{k!}\varphi^{(k)}(0)y^k+\frac{1}{(N+1)!}\varphi^{(N+1)}(\mathrm{y})y^{N+1}. \tag{7.11}$$

比较(2.12)可知，将 y 看成离散化参数 h，将 x 看成 $T(h)$，则 $x=\varphi(y)$具有(2.12)型的渐近展开式，其中 $r=1$. 因此，对任一种方法(如二分法、弦位法、牛顿法及其它迭代法)求出的数列 $\{x_n\}$ 及相应函数值数列 $\{y_n\}$，我们都可用多项式外推法或 Stoer 外推法来加速 $\{x_n\}$ 的收敛. 多项式外推法的计算公式，根据(2.13)、(2.14)，是

$$\left.\begin{aligned} &T_0^{(i)}=x_i,\\ &T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{y_i/y_{i+m}-1}. \end{aligned}\right\} \tag{7.12}$$

而 Stoer 外推法的计算公式，根据(3.34)、(3.33)，是

$$\left.\begin{aligned} &T_{-1}^{(i)}=0, \quad T_0^{(i)}=x_i, \\ &T_m^{(i)}=T_{m-1}^{(i+1)}+\frac{T_{m-1}^{(i+1)}-T_{m-1}^{(i)}}{\theta_m^{(i)}y_i/y_{i+m}-1}, \\ &\theta_m^{(i)}=\frac{T_{m-1}^{(i)}-T_{m-2}^{(i+1)}}{T_{m-1}^{(i+1)}-T_{m-2}^{(i+1)}}. \end{aligned}\right\} \tag{7.13}$$

数列 $\{x_n\}$ 可根据一种迭代法独立产生，也可利用外推过程中得到的 $T_m^{(i)}$ 产生. 例如采用牛顿迭代法时，可令

$$x_{n+1}=x_n-f(x_n)/f'(x_n), \tag{7.14}$$

也可令

$$x_{n+1}=T_n^{(0)}-f(T_n^{(0)})/f'(T_n^{(0)}). \tag{7.15}$$

以如下具体方程为例:

$$x^3-x-1=0. \tag{7.16}$$

取 $x_0=1$，按牛顿法(7.14)得

$$x_0=1, \quad x_1=1.5,$$

$$x_2=1.347826, \quad x_3=1.3252003,$$

$$x_4=1.3247181, \quad x_5=x_6=1.3247179.$$

如按(7.15)和(7.12)，则得表 7.7. 注意其中

$$x_3\approx1.3247180$$

已达到单纯牛顿法 x_4 的精度.

表 7.7

y_i	$T_0^{(i)}=x_i$		
−1	1		
0.875	1.5	1.2666667	
0.0145861	1.3281274	1.3252138	1.3243722
0	1.3247180		

§4 代数方程组的求解

设有线性代数方程组

$$Bs=b, \tag{7.17}$$

其中 B 是已知的 $p\times p$ 矩阵，b 是已知的 p 维向量，s 是未知的 p 维向量．为了求解 s，常常采用迭代法．令 $A=I-B$，则方程(7.17)变为

$$s=As+b. \tag{7.18}$$

任选 s_0，令

$$s_{j+1}=As_j+b,\quad j=0,\ 1,\ 2,\ \cdots. \tag{7.19}$$

则得一向量序列．对于这序列，我们可用 ε 算法加速其收敛；而且可以证明，利用有限个 s_j，即用有限次迭代，便可得到方程组(7.17)或(7.18)的准确解．

事实上，在矩阵 B 非奇异的情况下，我们有

$$\begin{aligned} s_j&=As_{j-1}+b=A^2s_{j-2}+Ab+b=\cdots\\ &=A^js_0+(I+A+A^2+\cdots+A^{j-1})b. \end{aligned}$$

但是
$$(I+A+A^2+\cdots+A^{j-1})(I-A)=I-A^j,$$

$$\begin{aligned} I+A+A^2+\cdots+A^{j-1}&=(I-A^j)(I-A)^{-1}\\ &=(I-A^j)B^{-1}, \end{aligned}$$

故有

$$s_j=A^js_0+(I-A^j)B^{-1}b=A^j(s_0-B^{-1}b)+B^{-1}b. \tag{7.20}$$

这样，设 A 关于 $s_0-B^{-1}b$ 的最小多项式为

$$\sum_{\nu=0}^{m}c_\nu\lambda^\nu, \tag{7.21}$$

即假定

$$\sum_{\nu=0}^{m} c_\nu A^\nu (\boldsymbol{s}_0 - B^{-1}\boldsymbol{b}) = 0, \tag{7.22}$$

则有
$$\sum_{\nu=0}^{m} c_\nu A^{\nu+j} (\boldsymbol{s}_0 - B^{-1}\boldsymbol{b}) = 0, \ j = 0, 1, 2, \cdots,$$

$$\sum_{\nu=0}^{m} C_\nu (S_{\nu+j} - B^{-1}b) = 0,$$

$$\sum_{\nu=0}^{m} c_\nu \boldsymbol{s}_{\nu+j} = B^{-1}\boldsymbol{b} \sum_{\nu=0}^{m} c_\nu. \tag{7.23}$$

于是，根据 ε 算法的性质 2，如果

$$\sum_{\nu=0}^{m} c_\nu \neq 0, \tag{7.24}$$

便有

$$\varepsilon_{2m}^{(i)} = B^{-1}\boldsymbol{b}. \tag{7.25}$$

即在 ε 表的第 $2m$ 列可得准确解.

条件(7.24)相当于最小多项式(7.21)没有根 $\lambda=1$. 在(7.24)不成立的情况下，

$$\sum_{\nu=0}^{m} c_\nu \lambda^\nu = (\lambda - 1)^\mu \sum_{\nu=0}^{m-\mu} c'_\nu \lambda^\nu,$$

$$(I - A)^\mu \sum_{\nu=0}^{m-\mu} c'_\nu A^\nu (\boldsymbol{s}_0 - B^{-1}\boldsymbol{b}) = 0,$$

$$\sum_{\nu=0}^{m-\mu} c'_\nu A^\nu (\boldsymbol{s}_0 - B^{-1}\boldsymbol{b}) = 0.$$

这跟(7.21)为最小多项式相矛盾. 所以，在最小多项式次数为m的假定下，条件(7.24)自然成立.

在 A 为奇异矩阵的情况下，最小多项式(7.21)中可能有

$$c_0 = c_1 = \cdots = c_\tau = 0.$$

此时，令 $c'_\nu = c_{\nu+\tau}$，则(7.22)变为

$$\sum_{\nu=0}^{m-\tau} c_\nu A^{\nu+\tau} (\boldsymbol{s}_0 - B^{-1}\boldsymbol{b}) = 0,$$

而(7.23)变为

$$\sum_{\nu=0}^{m-\tau} c'_\nu \boldsymbol{s}_{\nu+\tau+j} = B^{-1}\boldsymbol{b} \sum_{\nu=0}^{m-\tau} c'_\nu. \qquad (7.23')$$

于是根据 ε 算法的性质 2 可知

$$\varepsilon_{2(m-\tau)}^{(\tau+j)} = B^{-1}\boldsymbol{b}. \qquad (7.25')$$

这表明，在 ε 表的第 $2(m-\tau)$ 列，也可能得到解 $B^{-1}\boldsymbol{b}$.

应当注意，上面的证明并没有假定 $\boldsymbol{s}_j$ 收敛．这说明，即使序列 $\boldsymbol{s}_j$ 发散，利用 ε 算法，在 ε 表的第 $2m$ 或 $2(m-\tau)$ 列仍可得到准确解 $B^{-1}\boldsymbol{b}$.

还可以证明，在 $B=I-A$ 为奇异矩阵的时候，只要(7.17)有解，利用 ε 算法也可得到解．事实上，设解为 $\boldsymbol{s}$，初始向量为 $\boldsymbol{s}_0$，B 关于 $\boldsymbol{s}_0-\boldsymbol{s}$ 的最小多项式为 m 次，以 0 为单根，没有 $\lambda=1$ 的根，则 A 关于 $\boldsymbol{s}_0-\boldsymbol{s}$ 的最小多项式可写成如下的形式

$$\sum_{\nu=0}^{m} c'_\nu \lambda^\nu = (1-\lambda)\sum_{\nu=0}^{m-1} r_\nu \lambda^\nu, \quad r = \sum_{\nu=0}^{m-1} r_\nu \neq 0. \qquad (7.26)$$

此时，尽管(7.20)不成立，但仍有

$$\boldsymbol{s}_j = A^j\boldsymbol{s}_0 + (I + A + A^2 + \cdots + A^{j-1})\boldsymbol{b}.$$

注意(7.17)有解，$\boldsymbol{b}=(I-A)\boldsymbol{s}$，便知

$$\begin{aligned}\boldsymbol{s}_j &= A^j\boldsymbol{s}_0 + (I + A + A^2 + \cdots + A^{j-1})(I-A)\boldsymbol{s}\\ &= A^j\boldsymbol{s}_0 + \boldsymbol{s} - A^j\boldsymbol{s} = A^j(\boldsymbol{s}_0 - \boldsymbol{s}) + \boldsymbol{s},\end{aligned}$$

从而 $$\boldsymbol{s}_j - \boldsymbol{s}_{j+1} = (I-A)A^j(\boldsymbol{s}_0 - \boldsymbol{s}).$$

由此，利用(7.26)得

$$\begin{aligned}&\sum_{\nu=0}^{m-1} r_\nu \boldsymbol{s}_{j+\nu} - \sum_{\nu=0}^{m-1} r_\nu \boldsymbol{s}_{j+\nu+1}\\ &\quad = \left\{(I-A)^j \sum_{\nu=0}^{m-1} r_\nu A^\nu\right\}(\boldsymbol{s}_0 - \boldsymbol{s}) = 0.\end{aligned}$$

这表明，左边的和式为常向量，

$$\sum_{\nu=0}^{m-1} r_\nu \boldsymbol{s}_{j+\nu} = r\boldsymbol{\varepsilon}. \tag{7.27}$$

于是，根据 ε 算法的性质 2，得到

$$\boldsymbol{\varepsilon}_{2m-2}^{(i)} = \boldsymbol{\varepsilon}, \quad i=0, 1, 2, \cdots. \tag{7.28}$$

这里 $\boldsymbol{\varepsilon}$ 实质上就是(7.17)的解. 理由如下：在(7.27)中令 $j=0$，得

$$\boldsymbol{\varepsilon} = r^{-1}\sum_{\nu=0}^{m-1} r_\nu \boldsymbol{s}_\nu.$$

故
$$\begin{aligned} B\boldsymbol{\varepsilon} &= (I-A)r^{-1}\sum_{\nu=0}^{m-1} r_\nu \boldsymbol{s}_\nu \\ &= r^{-1}(I-A)\sum_{\nu=0}^{m-1} r_\nu \{A^\nu(\boldsymbol{s}_0-\boldsymbol{s})+\boldsymbol{s}\} \\ &= r^{-1}(I-A)\sum_{\nu=0}^{m-1} r_\nu \boldsymbol{s} = (I-A)\boldsymbol{s} = B\boldsymbol{s} = \boldsymbol{b}. \end{aligned}$$

如果 B 关于 $\boldsymbol{s}_0-\boldsymbol{s}$ 的最小多项式有 τ 重根 $\lambda=1$，上面由(7.26)开始的推导只要将m改为 $m-\tau$ 就仍然有效. 因此，

$$\boldsymbol{\varepsilon}_{2(m-\tau)-\tau}^{(i)} = \boldsymbol{\varepsilon}, \quad i=0, 1, 2, \cdots. \tag{7.28'}$$

这说明，在 B 为奇异矩阵的时候，仍可用 ε 算法得到(7.17)的解.

对于非线性方程组

$$\boldsymbol{s} = F(\boldsymbol{s}) \tag{7.29}$$

的迭代法
$$\boldsymbol{s}_{j+1} = F(\boldsymbol{s}_j),$$

我们也可应用 ε 算法来加速收敛. 这时尽管通常不能经过有限步迭代得出方程(7.29)的解，但可以得到二阶收敛于解的序列 $\boldsymbol{V}_n$. $\boldsymbol{V}_n$ 的构造方法如下：在(7.29) 的解 $\boldsymbol{V}$ 的邻域 D 上任取 $\boldsymbol{V}_0$. 对于 $n=0, 1, 2, \cdots$,

(1) 令 $\boldsymbol{s}_0 = \boldsymbol{V}_n$;

(2) 令 $\boldsymbol{s}_{j+1} = F(\boldsymbol{s}_j)$, $j=0, 1, \cdots, 2m+\tau-1$,

这里m是$F'(\boldsymbol{V})$($F(\boldsymbol{V})$在$\boldsymbol{V}$处的Frechet导数)关于$\boldsymbol{V}_n-\boldsymbol{V}$的最小多项式次数，$\tau$是这多项式的根$\lambda=0$的重数. 这里还假定$I-F'(\boldsymbol{V})$非奇异.

(3) 利用ε算法计算$\boldsymbol{V}_{n+1}=\varepsilon_{2(m-\tau)}^{(\tau)}$.

现在我们来证明$\boldsymbol{V}_n$二阶收敛于解$\boldsymbol{V}$. 先设$\tau=0$. 此时由于$F(\boldsymbol{V})$可导,

$$\begin{aligned}\boldsymbol{s}_{j+1}-\boldsymbol{V}&=F(\boldsymbol{s}_j)-F(\boldsymbol{V})\\&=F'(\boldsymbol{V})(\boldsymbol{s}_j-\boldsymbol{V})+o(\|\boldsymbol{s}_j-\boldsymbol{V}\|^2)\\&=\{F'(\boldsymbol{V})\}^{j+1}(\boldsymbol{s}_0-\boldsymbol{V})+o(\|\boldsymbol{s}_0-\boldsymbol{V}\|^2).\end{aligned}$$

设$F'(\boldsymbol{V})$关于$\boldsymbol{s}_0-\boldsymbol{V}$的最小多项式为(7.21),则

$$\begin{aligned}\sum_{\nu=0}^{m}c_\nu(\boldsymbol{s}_{j+\nu}-\boldsymbol{V})&=\{F'(\boldsymbol{V})\}^j\left\{\sum_{\nu=0}^{m}c_\nu[F'(\boldsymbol{V})]^\nu\right\}(\boldsymbol{s}_0-\boldsymbol{V})\\&\quad+\sum_{\nu=0}^{m}c_\nu o(\|\boldsymbol{s}_0-\boldsymbol{V}\|^2)\\&=\sum_{\nu=0}^{m}c_\nu o(\|\boldsymbol{V}_n-\boldsymbol{V}\|^2),\end{aligned}$$

从而

$$\sum_{\nu=0}^{m}c_\nu(\boldsymbol{s}_{j+\nu}-\boldsymbol{V}-o(\|\boldsymbol{V}_n-\boldsymbol{V}\|^2))=0.$$

于是根据ε算法的性质2,

$$\boldsymbol{V}_{n+1}=\varepsilon_{2m}^{(0)}=\boldsymbol{V}+o(\|\boldsymbol{V}_n-\boldsymbol{V}\|^2).$$

在$\tau\neq0$的情形亦可类似地证明.

应当注意，上面的证明并没有假定F为压缩映射，没有假定序列$\{s_j\}$收敛. 因此,应用ε算法的方程组(7.29),其范围很广泛. 在应用ε算法时，不必求$F'(\boldsymbol{V})$的逆，而所得序列又二阶收敛,可见这种加速法十分有效.

微分方程、积分方程、最优化问题常常归结为线性或非线性代数方程组的求解. ε算法自然可应用于这些问题的求解.

参考文献

全书主要参考文献

Joyce, D. C., Survey of extrapolation processes in numerical analysis, SIAM Rev. 13(1971), 435—490(译文见西安交通大学科技参考资料74-028号)

第1章

[1] T. E. Hull 等，常微分方程数值方法的比较，计算机应用与应用数学，1975, 3.

[2] 钱宝琮，中国数学史，科学出版社，1964.

[3] 自然科学史研究所，中国古代科技成就，中国青年出版社，1978.

[4] 李约瑟，中国科学技术史，第三卷，科学出版社，1978.

[5] 中国科学院沈阳计算技术研究所等，电子计算机常用算法，科学出版社，1976.

第2章

[1] 李岳生、黄友谦，数值逼近，人民教育出版社，1978.

[2] 南京大学数学系计算数学专业，数值逼近方法，科学出版社，1978.

[3] R. Bulirsch, J. Stoer, Fehlerabschätzungen und extrapolation mit rationalen Funktionen bei Verfahren Vom Richardson Typus, Numer. Math. 6(1964), 413～427(译文见西安交通大学科技参考资料74-028号)

[4] H. E. Salzer, A simple method for summing certain slowly convergent series, J. Math. and Phys. 33(1954), 356～359.

[5] G. Mühlbach, A Recurrence Folmula for Generalized Divided Differences and some Applications, J. Appro. Theory, 9(1973), 165～172.

[6] G. Mühlbach, Neville-Aitken Algorithms for Interpolation by Functions of Čebyšev-Systems in the Sense of Newton and in a generalized Sense of Hermite, 见 A. G. Law and B. N. Sahney 编 Theory of Approximation with Applications, AP, 1976, 200～212.

[7] G. Mühlbach, Error Estimates for Extrapolation Operators, J. Approximation Theory, 21(1977), 328～332.

[8] G. Mühlbach, The general Neville-AitkenAlgorithm and some applications, Numer. Math. 31(1978), 97～110.

[9] R. Bulirsch and J. Stoer, Asymptotic upper and lower bounds for results of extrapolation methods, Numer. Math. 8(1966), 93～104.

[10] K. Knopp, Theory and Application of Infinite Series, Blackie and Son Limited, 1957, 74～75.

第3章

[1]～[3] 同第2章[1]、[3]、[9].

[4] L. Wuytack, An Algorithm for Rational Interpolation similar to the qd-Algorithm, Numer. Math. 20(1973), 418—424.

[5] F. M. Larkin, Some Techniques for Rational Interpolation, Computer J. 10(1967), 178～186.

[6] L. Wuytack, A New Technique for Rational Extrapolation to the Limit, Numer. Math. 17(1971), 215～221.

[7] 星野聡,数值计算の技法,コロナ社,昭和52年.

第4章

[1] 同第3章[7]

[2] P. Wynn, On Divice for Computing the $e_m(S_n)$ Transformation, Math. Comp. 10(1956), 91～96.

[3] A. C. Aitken, Determinants and Matrices, Oliver and Boyd, 1958, 103～108.

[4] L. Brezinski, Computation of the Eigenelements of a Matrix by the ε-Algorithm, Linear Algebra, 11(1975), 7～20.

[5] P. Wynn, Acceleration Technique for Iterated vector and Matrix Problems, Math. Comput. 16(1962), 301～322.

[6] D. Shanks, Non-Linear Transformations of Divergent and Slowly Convergent Sequence, J. Math. and Phys. 34(1955), 1～42.

[7] A. C. Genz, Application of ε-Algorithm to Quadrature problems,见 R. Graves-Morris 编 Padé approximants and their applications, AP, 1973, 105～116.

第5章

[1] 伊·彼·梅索夫斯基赫,计算方法,人民教育出版社, 1960.

[2] E. T. Whittaker and G. N. Watson, A Course of Modern Analysis, 1952.

[3] R. Bulirsch, Bemerkungen zur Romberg-Integration. Numer. Math.

6(1964), 6～16(译文见西安交通大学科技参考资料 74-023 号).
[4] 清华大学、北京大学,计算方法(上册),科学出版社, 1974.
[5] T. Havie, On a modification of Romberg's Algorithm, BIT6 (1966), 24～30.
[6] P. J. Davis and P. Rabinowitz, Methods of Numerical Integration, AP, 1975.
[7] 同第4章[7]
[8] G. M. Phillips and P. J. Taylor, Theory and Applications of Numerical Analysis, AP, 1973. (有中译本,西安交通大学译印, 1977).
[9] A. M. Cohen 等, Numerical Analysis, McGraw-Hill, 1973 (有中译本, 西安交通大学译印, 1976).
[10] T. Havie, On the Practical Application of the Modified Romberg Algorithm, BIT7 (1967), 103～113.
[11] L. Fox, Romberg Integration for a Class of singular Integrands, Computer J. 10(1967), 87～93.
[12] J. N. Lyness and B. W. Ninham, Numerical Quadrature and Asymptotic Expansions, Math. Comp. 21(1967), 162～178.
[13] J. N. Lyness and C. B. Moler, Vandermonde Systems and Numerical Differentiation, Numer. Math. 8(1966), 458～464.
[14] J. N. Lyness and C. B. Moler, Generalized Romberg Method for Integration of Derivatives, Numer. Math. 14(1969), 1～13.

第6章

[1] L. Fox, Numerical Solution of Ordinary and Partial Differential Equations, Pergamon Press, 1962.
[2] P. Henrici, Disrete Viariable Methods in Ordinary Differential Equations, John Wiley and Sons, 1962.
[3] J. D. Lambert, Computatinal Methods in Ordinary Differential Equations, John Wiley and Sons, 1973.
[4] L. Lapidus and J. H. Seinpeld, Numerical Solution of Ordinary Differential Equations, AP. 1971.
[5] C. W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, 1971.
[6] G. Hall and J. M. Watt, Modern Numerical Methods for Ordinary Differential Equations, Clarendon Press, 1976.
[7] 南京大学数学系计算数学专业,偏微分方程数值解法,科学出版社, 1979.

[8] H. J. Stetter, Asymptotic Expansions for the Error of Discretization Algorithms for Non-linear Functional Equatious, Numer. Math. 7(1965), 18～31.

[9] A. H. 柯莫果洛夫、C. B. 佛明，函数论与泛函分析初步，高等教育出版社，1957.

[10] J. M. Drtega, W. C. Rheinboldt，多变量非线性方程组的迭代解法，游兆永译，西安交通大学印，1978.

[11] H. J. Stetter, Symmetric Two-Step Algorithms for Ordinary Differential Equations, Computing 5(1970), 267～280.

[12] 同第1章[1].

第7章

[1] 同第3章[4].

[2] 同第5章[8].

[3] C. Brezinski, A. C. Rien, The Solution of Systems of Equations Using the ε–Algorithm, and an Application to Boundary-Value Problems, Math. Comp.28(1974), 731～741.

算法索引

名　称　　页数

Romberg 外推算法 ……………………4
Richardson 外推法 ……………………8
h^2 外推、(h^2、h^4)外推……………………8
多项式外推法 ……………………18
Neville 外推算法 ……………………18
Bulirsch-Stoer 外推法 ……………………18
Lagrange 外推算法……………………23
Mühlbach 外推算法……………………28
Stoer 连分式外推法……………………49
Thiele 算法 ……………………50
Wuytack 算法 ……………………50
Larkin 算法 ……………………57
Stoer 有理式外推算法……………………57
Bulirsch-Stoer 有理式外推算法……………………57
ρ 算法 ……………………58
倒差商算法 ……………………58
带权的 ε 算法 ……………………58
Aitken Δ^2 加速法……………………69
ε 算法 ……………………69
Romberg 积分法 ……………………98
Lyness-Moler 外推算法……………………114
GBS 算法 ……………………143

[General Information]
书名=外推法及其应用
作者=邓建中
页数=161
SS号=10070524
DX号=
出版日期=1984年03月第1版
出版社=上海科学技术出版社

封面页
书名页
版权页
前言页
目录页
第1章　什么叫外推法
　　1　数学史上的一个疑案
　　2　Romberg 外推算法
　　3　Romberg 算法的理论根据
　　4　外推算法在计算机上的实现
　　5　外推法研究的问题
第2章　多项式外推法
　　1　多项式插值法基础
　　2　多项式外推法及其推广
　　3　广义多项式外推法
　　4　广义多项式插值法与 Richardson 外推法的关系
　　5　误差估计
　　6　收敛性与稳定性
第3章　有理式外推法
　　1　有理式插值的概念
　　2　连分式算法
　　3　Larkin 逐步插值法
　　4　有理式逐步外推法
　　5　误差估计
　　6　收敛性与稳定性
第4章　?算法
　　1　?算法的导出
　　2　?算法的若干性质
第5章　外推法的应用(一)：数值积分与微分

1 Euler-Maclaurin 求和公式
2 一些简单求积公式的导出
3 Romberg 积分法及其改进
4 反常积分
5 重积分
6 数值微分
第6章 外推法的应用(二)：函数方程数值解
1 Euler 折线法与外推
2 离散化方法与外推
3 全程误差渐近展开式
4 GBS 算法
第7章 外推法的应用(三):其它
1 加速序列的收敛
2 级数求和与特殊函数的求值
3 方程求根
4 代数方程组的求解
参考文献
算法索引
附录页